

## 02 HTML (HyperText Markup Language)

Einführung in die moderne (Full-Stack) Webentwicklung mit  
JavaScript, Node.js und React.js

Nils Constantin Hellwig  
Wissenschaftliche Hilfskraft  
**RECHENZENTRUM**



Universität Regensburg

# Überblick Themen

- Was ist HTML?
- Aufbau eines HTML-Dokuments
- Links
- Bilder einbinden und formatieren
- Text
- Formulare
- Listen

## Was ist HTML?

- HTML steht für HyperText Markup Language
- HTML beschreibt die Struktur einer Website
- HTML besteht aus Tags, die verwendet werden, um die Struktur und den Inhalt einer Webseite zu definieren
- Eine HTML-Datei besitzt die Endung .html oder .htm

## HTML-Dokumente im Browser anzeigen lassen

- Der HTML-Code kann in einem Browser geöffnet werden, indem man die Datei über den Dateimanager öffnet
- Um eine Website auch für andere Nutzer\*innen zur Verfügung zu stellen, bietet es sich an, den HTML-Code auf einen Server hochzuladen und damit bereitzustellen
- Besitzt man keinen eigenen Server, gibt es viele Tools (z.B. "ngrok", "http-server" für Node.js; z.B. "http-server" - <https://gist.github.com/willurd/5720255> ), die es ermöglichen, eine lokale Datei zu hosten, um darauf zugreifen zu können

## HTML: index.html

- Die Datei index.html ist die Standard-Startseite eines Webserver
- Wird die URL einer Website (z.B. [www.uni-regensburg.de](http://www.uni-regensburg.de)) aufgerufen, kann die index.html - Datei in der Kommunikation zwischen Client und Server automatisiert als Startseite identifiziert werden

# HTML-Elemente

- Ein HTML-Element ist durch ein Start-Tag, einen Inhalt und ein End-Tag definiert.

```
<tagname>Der Inhalt steht hier...</tagname>
```

- HTML-Elemente können verschachtelt werden (d.h. Elemente können andere Elemente enthalten)
- Alle HTML-Dokumente bestehen aus verschachtelten HTML-Elementen
- HTML unterscheidet nicht zwischen Groß- und Kleinschreibung (z.B. `<P>` und `<p>` bedeuten dasselbe)
- Das W3C (*World Wide Web Consortium*) empfiehlt jedoch Kleinschreibung für die Benennung von HTML-Tags

## Aufbau eines HTML-Dokuments

- Alle HTML-Dokumente sollte mit einer Dokumenttyp-Deklaration beginnen:

`<!DOCTYPE>`

Es dient dazu, die Art des Dokumentes an den Browser zu übermitteln, damit der Browser weiß, wie das Dokument interpretiert werden sollte. Die Deklaration für HTML5 ist `<!DOCTYPE html>`.

- Das HTML-Dokument selbst beginnt mit `<html>` und endet mit `</html>`
- Der sichtbare Teil des HTML-Dokuments befindet sich im `<body>` - Element

## Aufbau eines HTML-Dokuments: Beispiel

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```



## HTML-Attribute

- HTML-Attribute liefern zusätzliche Informationen über HTML-Elemente
- Alle HTML-Elemente können Attribute haben
- Attribute werden immer im Start-Tag angegeben
- Attribute bestehen normalerweise aus Name/Wert-Paaren wie: `name="Wert"`

## HTML-Links

- Links ermöglichen es den Nutzern, sich von Seite zu Seite zu klicken
- Der HTML `<a>` - Tag definiert einen Hyperlink. Er hat die folgende Syntax:

```
<a href="url...">link text</a>
```

- Das `href`-Attribut gibt das Ziel des Links an
- Beispiel:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>HTML-Link Beispiel</h1>
    <a href="https://www.uni-regensburg.de/">Website der Universität Regensburg</a>
  </body>
</html>
```

### HTML-Link Beispiel

[Website der Universität Regensburg](https://www.uni-regensburg.de/)

## HTML-Links: Relative Pfade

- Ein lokaler Link (ein Link zu einer Seite innerhalb derselben Website) wird mit einer relativen URL angegeben
- Beispiele:

```
<a href="impressum.html">Impressum</a>
```

```
<a href="/about">Über Uns</a>
```

## HTML-Links: Andere Elemente mit einem Link versehen

```
<a href="https://www.uni-regensburg.de">  
    
</a>
```

In diesem Beispiel wird das `<div>` - Element von einem `<a>` - Element umgeben, das das `href`-Attribut mit der URL `"https://www.uni-regensburg.de"` enthält. Dadurch wird das gesamte `<div>` - Element zu einem Link, der auf diese URL verweist, wenn man darauf klickt.

## HTML-Links: Das target-Attribut

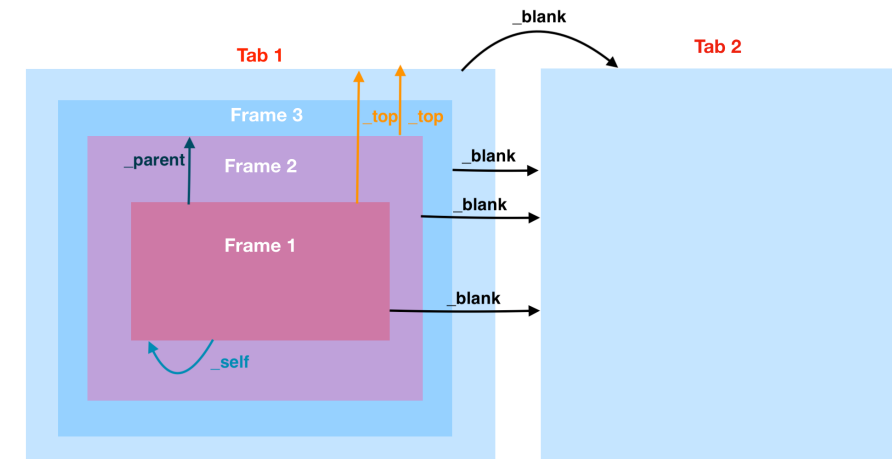
- Standardmäßig wird die verlinkte Seite im aktuellen Browserfenster angezeigt
- Um dies zu ändern, muss man ein anderer Wert für das `target` - Attribut des Links spezifiziert werden

`_self` – Öffnet den Link im selben Tab

`_blank` – Öffnet den Link in einem neuen Tab

`_parent` – Öffnet den Link in seinem parent frame

`_top` – Öffnet den Link im aktuellen Tab in dem das frame ist



## HTML-Images: Attribute

- Der `<img>` - Tag wird verwendet, um ein Bild in einer HTML-Seite einzubetten
- Das `src`-Attribut gibt den Pfad zu dem anzuzeigenden Bild an
- Die Attribute `width` und `height` dienen dazu, die Breite und Höhe des Bildes (in Pixeln) anzugeben
- Das `alt` - Attribut gibt einen alternativen Text für ein Bild an, wenn das Bild nicht angezeigt werden kann
- Beispiel:

```

```

## HTML-Images: Icons



- Das `<img>` - Element kann auch verwendet werden, um Icons darzustellen
- Für Icons sollte man möglichst Vektorgrafiken (z.B. im .svg-Format) verwenden
- Anders als Pixelgrafiken bestehen Vektorgrafiken nicht aus einzelnen Bildpunkten, sondern aus sogenannten grafischen Primitiven wie Linien, Kreisen oder Kurven, die durch Parameter wie Anfangspunkt, Endpunkt, Radius, Kantenlänge, Linienstärke, Farbe und Füllmuster definiert werden
- Skaliert man Rastergrafiken (z.B. .jpg oder .png) aus Pixeln, werden die Pixel größer und es kann zu Qualitätsverlusten kommen
- Umfangreiche Bibliothek mit Icons (auch im .svg-Format): <https://phosphoricons.com/>

# HTML-Headings

- HTML-Headings sind Überschriften und Unterüberschriften
- Beispiel:

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6



## HTML-Headings

- Suchmaschinen verwenden die Überschriften, um die Struktur und den Inhalt Ihrer Webseiten zu indizieren.
- Benutzer\*innen überfliegen eine Seite oft anhand der Überschriften
- `<h1>` - Überschriften sollten für die Hauptüberschriften verwendet werden, gefolgt von `<h2>` - Unterüberschriften usw.

## HTML-Paragraphs

- Das HTML-Element `<p>` definiert einen Paragraphen
- Ein Absatz beginnt immer in einer neuen Zeile, und die Browser fügen vor und nach einem Absatz automatisch etwas „margin“ ein (bei den meisten Browsern nach oben und unten 16px)

## HTML-Paragraphs

- Bei HTML können Sie die Abstände zwischen Textzeilen nicht ändern, indem Sie zusätzliche Leerzeichen oder Zeilen in Ihren HTML-Code einfügen
- Der Browser entfernt automatisch alle zusätzlichen Leerzeichen und Zeilen, wenn die Seite angezeigt wird:

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_paragraphs2](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_paragraphs2)

## HTML-Paragraphs: Line Breaks

- Das HTML-Element `<br>` definiert einen Zeilenumbruch:

```
<!DOCTYPE html>
<html>
  <body>
    <p>This is<br/>a paragraph<br/>with line breaks.</p>
  </body>
</html>
```

This is  
a paragraph  
with line breaks.

## HTML-Paragraphs: Weitere Textformatierungen

- `<b>` - Fetter Text
- `<strong>` - semantisch sehr bedeutsamen Text hervorheben, im Browser wird Text innerhalb dieses Tags fett dargestellt (relevant v.a. für Suchmaschinen)
- `<i>` - Kursiver Text
- `<em>` - Kursiver Text, semantische Bedeutung: Bedeutung hervorheben
- `<mark>` - Markierter Text (standardmäßig in Gelb)
- `<del>` - Durchgestrichener Text

## HTML-Forms

- Ein HTML-Formular wird verwendet, um Nutzereingaben zu erfassen
- Nutzereingaben werden in den meisten Fällen zur Verarbeitung an einen Server gesendet
- Das HTML-Element `<form>` wird verwendet, um ein HTML-Formular zu erstellen
- Das `<form>` - Element ist ein Container für verschiedene Arten von Eingabeelementen, wie z.B.: Textfelder, Radio-Buttons, Optionsfelder, Sendeknöpfe

```
<form>  
...  
form elements  
...  
</form>
```

## HTML-Forms: `<input>` - Element

- Das `<input>` - Element erlaubt Nutzereingaben in einem Formular
- Ein `<input>` - Element kann je nach Typ-Attribut auf verschiedene Weise dargestellt werden

`<input type="text">` - Texteingabefeld

☒ CSS

`<input type="radio">` - Radio-Button (z.B. zur Auswahl einer Möglichkeit)

☐ JavaScript

`<input type="checkbox">` - Kontrollkästchen (zur Auswahl von null oder mehr von mehreren Möglichkeiten)

Submit

`<input type="submit">` - Schaltfläche zum Senden (des Formulars)

Button

`<input type="button">` - Klickbare Schaltfläche

## HTML-Forms: `<label>` - Element

- Das `<label>` - Element in HTML dient dazu, eine Bezeichnung für ein `<form>` - Element bereitzustellen
- Wenn man ein `<label>` - Element hinzufügt, kann man das zugehörige Formularelement per Klick auf die Beschriftung auswählen
- Beispiel: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_form\\_radio](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_radio)



## HTML-Forms: Der Submit-Button

- Das `action` - Attribut in einem Formular-Element gibt die URL oder den Pfad an, an die die Formulardaten gesendet werden sollen, wenn der Submit-Button geklickt wird

Beispiel:

First name:

Last name:

```
<form action="/api/users">
  <label for="fname">First name:</label><br />
  <input type="text" id="fname" name="fname" value="John" /><br />
  <label for="lname">Last name:</label><br />
  <input type="text" id="lname" name="lname" value="Doe" /><br /><br />
  <input type="submit" value="Submit" />
</form>
```

- Standardmäßig werden die Daten mittels eines GET-Request übermittelt (im body)
- Die Methode kann jedoch geändert werden, anhand des Attributs `method` des `<form>` - Elements: `<form action="/api/users" method="POST">`

## HTML-Lists

| Unordered List   | Ordered List   | Description List   |
|--|--|--|
| <pre>&lt;ul&gt;   &lt;li&gt;Coffee&lt;/li&gt;   &lt;li&gt;Tea&lt;/li&gt;   &lt;li&gt;Milk&lt;/li&gt; &lt;/ul&gt;</pre> | <pre>&lt;ol&gt;   &lt;li&gt;Coffee&lt;/li&gt;   &lt;li&gt;Tea&lt;/li&gt;   &lt;li&gt;Milk&lt;/li&gt; &lt;/ol&gt;</pre> | <pre>&lt;dl&gt;   &lt;dt&gt;Coffee&lt;/dt&gt;   &lt;dd&gt;hot&lt;/dd&gt;   &lt;dt&gt;Milk&lt;/dt&gt;   &lt;dd&gt;cold&lt;/dd&gt; &lt;/dl&gt;</pre> |
| <ul style="list-style-type: none"> <li>• Coffee</li> <li>• Tea</li> <li>• Milk</li> </ul>                              | <ol style="list-style-type: none"> <li>1. Coffee</li> <li>2. Tea</li> <li>3. Milk</li> </ol>                           | <div>Coffee</div> <div>hot</div> <div>Milk</div> <div>cold</div>   |
| "li" steht für "list item"   |  | "dt" steht für "definition term"<br>"dd" steht für "definition description"  |

## HTML-Head

- Das `<head>` - Element ist ein Container für Metadaten (Daten über Daten) und befindet sich zwischen dem `<html>` - Tag und dem `<body>` - Tag
- HTML-Metadaten sind Daten über das HTML-Dokument, sie werden nicht im Viewport angezeigt
- Metadaten definieren in der Regel den Titel des Dokuments, den Zeichensatz, Stile, Skripte und andere Metainformationen

## HTML-Head: `<title>` - Element

- Das `<title>` - Element definiert den Titel des Dokuments
- Der Titel muss ein reiner Text sein und wird in der Titelleiste des Browsers angezeigt
- Beispiel:

```
<head>
```

```
  <title>To-Do App</title>
```

```
<head>
```



## HTML-Head: `<link>` - Element

- Das `<link>` - Element wird in HTML verwendet, um eine externe Ressource, wie z.B. eine CSS-Datei oder einen RSS-Feed, auf einer HTML-Seite zu verlinken
- Der `<link>` - Tag wird am häufigsten für Links zu externen StyleSheets verwendet
- Beispiel:

```
<head>
```

```
    <link rel="stylesheet" href="style.css" />
```

```
</head>
```

## HTML-Head: `<meta>` - Element

- Das `<meta>` - Element wird in der Regel verwendet, um den Zeichensatz, die Seitenbeschreibung, die Schlüsselwörter, den Autor des Dokuments und die Ansichtseinstellungen anzugeben
- Die Metadaten werden vor allem von Suchmaschinen (Schlüsselwörter) verwendet

## HTML-Head: `<meta>` - Element

| Character Set   | Author  |
|---|---|
| <code>&lt;meta charset="UTF-8"&gt;</code>                                 | <code>&lt;meta name="author" content="John Doe"&gt;</code>                                |
| Keywords  | Page Refresh  |
| <code>&lt;meta name="keywords" content="HTML, CSS, JavaScript"&gt;</code> | <code>&lt;meta http-equiv="refresh" content="30"&gt;</code>                               |
| Description   | Viewport  |
| <code>&lt;meta name="description" content="Cats and Dogs"&gt;</code>      | <code>&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;</code> |

## HTML-Head: viewport

- Der Viewport ist der für Benutzer\*innen sichtbare Bereich einer Webseite
- Sie sollten das folgende `<meta>` - Element in alle Ihre Webseiten aufnehmen:  
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

- `width=device-width`


Legt die Breite der Seite so fest, dass sie der Bildschirmbreite des Geräts entspricht (die je nach Gerät variiert)

- `initial-scale=1.0`

Legt die anfängliche Zoomstufe fest, wenn die Seite zum ersten Mal vom Browser geladen wird



## HTML-Head: Favicon

 To-Do App

- Ein Favicon ist ein kleines Bild, das neben dem Seitentitel in der Browser-Registerkarte angezeigt wird

- Beispiel:

```
<head>
```

```
  <title>To-Do App</title>
```

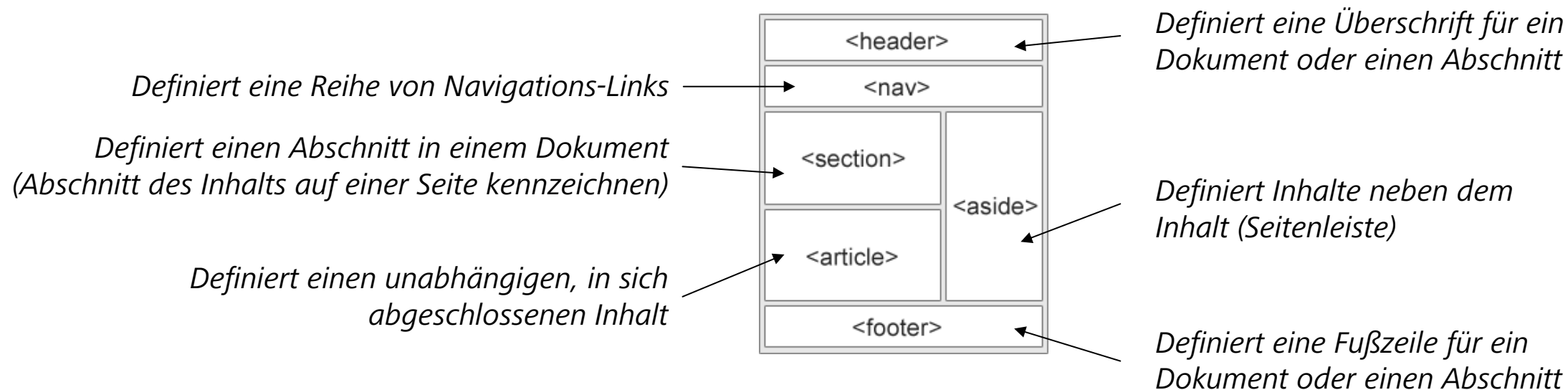
```
  <link rel="icon" type="image/x-icon" href="/icons/favicon.ico"/>
```

```
</head>
```

- `image/x-icon` gibt den Medientyp des Icons an (auch z.B. `image/png`)

# HTML Layout Elemente

- HTML hat mehrere semantische Elemente, die die verschiedenen Teile einer Webseite definieren:



## HTML div-Container

- div-Container sind HTML-Elemente, die dazu verwendet werden, eine Gruppe von Inhalten zusammenzufassen und zu organisieren
- Durch die Verwendung von div-Containern können komplexe Layouts erstellt werden und die Positionierung und Ausrichtung von Elementen auf einer Webseite genau steuern

```
<div class="title-section">  
  <h2>This is a heading in a div element</h2>  
  <p>This is some text in a div element.</p>  
</div>
```

## HTML Klassen

- Das HTML-Attribut `class` wird verwendet, um eine Klasse für ein HTML-Element anzugeben
- Mehrere HTML-Elemente können dieselbe Klasse haben
- Das `class` - Attribut wird häufig verwendet, um auf einen Klassennamen in einem Stylesheet zu verweisen
- Beispiel: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_classes\\_capitals](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_classes_capitals)

## HTML Id

- Das `id` – Attribut ist ein eindeutiger Identifikator für ein HTML-Element
- Der Wert des Attributs `id` muss innerhalb des HTML-Dokuments eindeutig sein
- Ein Klassenname kann von mehreren HTML-Elementen verwendet werden, während ein `id`-Name nur von einem HTML-Element innerhalb der Seite verwendet werden darf
- Das Attribut wird verwendet, um auf eine bestimmte Stildeklaration in einem Stylesheet zu verweisen
- Beispiel: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_id\\_class](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_id_class)

## HTML Naming Conventions: Ids&Klassen

- Wenn die Id oder Klasse aus einem einzigen Wort (oder einer Abkürzung) besteht, verwendet man Kleinbuchstaben, z.B. `"card"`
- Wenn die Id oder Klasse aus zwei oder mehr Wörtern besteht, verwendet man ausschließlich Kleinbuchstaben und trennt benachbarte Wortpaare mit einem Bindestrich, z.B. `"card-list"` oder `"food-items"`

## HTML-Comments

- HTML-Kommentare werden im Browser nicht angezeigt, aber sie können helfen, HTML-Quellcode zu dokumentieren

- Beispiel:

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

- Man kann auch mehr als eine Zeile auskommentieren:

```
<!--
```

```
<p>Look at this image:</p>
```

```

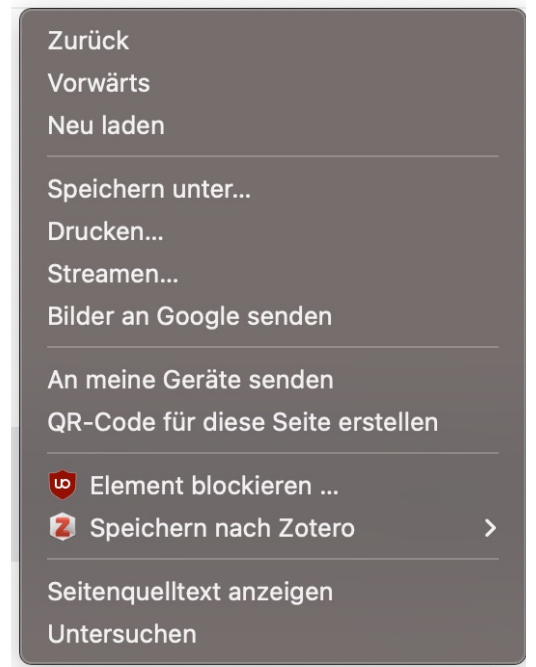
```

```
-->
```

```
<p>This is a paragraph too.</p>
```

## HTML Source Code einer Website im Browser betrachten

- Mit einem Rechtsklick auf eine Seite im Browser und durch einen Klick auf „Seiten Quelltext anzeigen“ (in Chrome) oder „Quelltext anzeigen“ (in Edge) kann man den Quellcode einer Website betrachten
- Durch einen Rechtsklick auf ein Element auf einer Seite im Browser und durch einen Klick auf „Untersuchen“ kann der Quellcode eines einzelnen Elements betrachtet und editiert werden sowie Manipulationen an CSS-Definitionen vorgenommen werden





## Weiterführende Literatur / Quellen

- Ausführliches Tutorial zu HTML mit anschaulichen Beispielen:  
[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- CS106E Stanford – Intro to HTML:  
<https://web.stanford.edu/class/cs106e/lectureNotes/L10NIntroHTML.pdf>