

# Erste Schritte mit HTML

Peter Willadt

LES Pforzheim

2021-03-08

# Was ist HTML?

HTML bedeutet *Hypertext Markup Language*

# Was ist HTML?

HTML bedeutet *Hypertext Markup Language*

Hypertext ist Text mit Links, die zu anderen Texten führen.

# Was ist HTML?

**HTML** bedeutet *Hypertext Markup Language*

**Hypertext** ist Text mit Links, die zu anderen Texten führen.

**Markup** ist die Kennzeichnung von Text.

# Was ist HTML?

**HTML** bedeutet *Hypertext Markup Language*

**Hypertext** ist Text mit Links, die zu anderen Texten führen.

**Markup** ist die Kennzeichnung von Text.

- HTML ist rein deklarativ, es ist *keine* Programmiersprache.

# Vorteile

- sehr weit verbreitet
- unabhängig von bestimmten Firmen und Betriebssystemen
- klar strukturiert
- einfach zu lernen
- einfach zu erzeugen
- einfach umzuwandeln

# HTML kommt selten allein

- HTML für semantischen Markup
- CSS für visuellen Markup
- JavaScript für Interaktion im Browser
- PHP für Interaktion auf dem Server

# Generieren oder selbst schreiben?

## Generieren

- Komfortable Arbeitsumgebung erleichtert Texterfassung.
- Tabellen, Formulare, Style-Sheets sind aufwendig zu schreiben.

# Generieren oder selbst schreiben?

## Generieren

- Komfortable Arbeitsumgebung erleichtert Texterfassung.
- Tabellen, Formulare, Style-Sheets sind aufwendig zu schreiben.

## Selbst schreiben

- Maschinell erstellter Code ist aufgebläht.
- Sauberer, kompakter Code lädt schnell und problemlos.
- Zusätzlicher semantischer Markup erleichtert Maschinen das Textverständnis.
- Zusätzlicher Markup fördert Barrierefreiheit.

# Werkzeugkasten

- Programmier-Editor
- Webbrowser
- HTML-Sprachreferenz
- Syntaxprüfer
- Link-Prüfer
- Sicherheits-Prüfer
- Webserver (für Fortgeschrittene)

# Programmier-Editoren: Features

- Syntax-Highlighting
- Wortergänzung
- Formatier-Hilfen
- Syntax-Check
- Makro-Fähigkeit

# Legen Sie los!

- Besorgen Sie sich Notepad++.
- Erstellen Sie eine einfache HTML-Datei.
- Schauen Sie die Datei im Browser Ihrer Wahl an.
- Verwenden Sie den Entwickler-Modus des Browsers.
- Bearbeiten Sie Ihre Datei, klicken Sie im Browser auf *Neu Laden*.

# HTML Nachschlagen

**Selfhtml** Ausführlich, solide, deutschsprachig. Auch für Einsteiger.

**MDN** (Mozilla Developer Network): vor allem für neueste Features wichtig. Nur teilweise deutsch.

**W3Schools** Mit vielen Beispielen, auch aus verwandten Gebieten. Englisch.

# Sprachelemente

## Tags `<html>`

- Tags sind das wesentliche Sprachelement.
- Tags sind in spitze Klammern eingeschlossen.
- Kleinschreibung wird empfohlen, Großschreibung funktioniert meist auch.
- Die meisten Tags treten paarweise auf. Das Paar umschließt den getaggten Text.

## Entities `&nbsp;`

- Entities stellen einzelne Zeichen dar.
- Entities beginnen mit einem & und enden mit einem Semikolon.
- Groß- und Kleinschreibung wird beachtet.

# Tags

- Schließende Tags unterscheiden sich von öffnenden Tags durch einen / am Anfang: `<h1>Überschrift</h1>`
- Manche Tags gibt es nicht in schließender Form. Man kann diesen einen / ans Ende stellen: `<br>` oder `<br />`
- Manche Tags nehmen Attribute (zusätzliche Angaben) auf: `<a href="Link-URL">Ein Link</a>`

# Arten von Tags

## Block-Tags (z.B. <body>, <h1>, <p>)

- Block-Tags kennzeichnen die Struktur der Seite: Überschriften, Absätze, Listen ...
- Block-Tags werden nur in eng begrenztem Umfang und nach strengen Regeln verschachtelt.

## Inline-Tags (z.B. <b>, <em>, <sub>)

- Inline-Tags markieren einzelne Text-Elemente: hervorgehobener Text, tiefgestellter Text, Link ...
- Sie dürfen innerhalb von Block-Tags verwendet werden und verschachtelt werden, aber sich nicht überlappen.

# Arten von Tags

**visuell** Diese Tags geben an, wie der Text dargestellt werden soll. Sie sollten nicht mehr verwendet werden.

```
<b>fett</b>
```

**semantisch** Diese Tags geben an, was der Text bedeutet.

```
<strong>wichtig</strong>
```

**Meta-Tags** Sie liefern Informationen *über* den Text:

```
<title>Seitentitel</title>
```

```
<meta name="author" content="Fred Beispiel">
```

Wenn Sie die Auswahl zwischen visuellen und semantischen Tags haben, verwenden Sie *semantische* Tags.

# Entities &nbsp;

- Entities stellen einzelne Zeichen dar.
- Typische Gründe:
  - Das Zeichen gibt es nicht auf der Tastatur.
  - Das Zeichen könnte den HTML-Browser verwirren (wie < oder &).
  - Im verwendeten Zeichensatz gibt es dieses Zeichen nicht.
- Die meisten Entities kann man sich sparen, wenn man UTF8-Kodierung für seinen Text verwendet. Das geht so:

```
<head>
```

```
<meta charset="utf-8"/>
```

```
...
```

# Beispiele für Entities

<i>Entity</i>	<i>Darstellung</i>
<code>&amp;lt;</code>	<
<code>&amp;gt;</code>	>
<code>&amp;amp;</code>	&
<code>&amp;nbsp;</code>	geschütztes Leerzeichen
<code>&amp;shy;</code>	weiches Trennzeichen
<code>&amp;Auml;</code>	Ä
<code>&amp;auml;</code>	ä
<code>&amp;#8623;</code>	Zeichen Nummer 8623 (dezimal)

# Attribute

- Attribute ergänzen Tags.
- Manche Attribute sind verpflichtend, andere optional.
- Attribute bestehen meist aus einem Schlüsselwort, einem Gleichheitszeichen und einem Wert in Anführungszeichen.

```

```

# CSS

- CSS regelt das Aussehen von HTML-Seiten.
- CSS ermöglicht durchgängige Formatierung eines Webauftritts.
- Mit CSS ist an das Medium angepasste Formatierung möglich (z.B. unterschiedlich für Smartphone, PC-Bildschirm, Drucker).
- CSS kann für Arten von Elementen wie auch für einzelne Elemente formuliert werden.

Die HTML-Tags `<div>` (Block) und `<span>` (Inline) und die Attribute *class* und *id* kennzeichnen Abschnitte für CSS oder JavaScript.