

Die Verwendung des `script`-Elements, um ein externes JavaScript mit dem `src`-Attribut zu referenzieren, will ich Ihnen nicht vorenthalten:

```
...
<head>
  <title>Das script-Element verwenden</title>
  <script type="text/javascript" src="script.js"></script>
...
</head>
...
```

**Listing 3.6** /Beispiele/Kapitel003/3\_7/index2.html

In diesem Beispiel wird davon ausgegangen, dass sich ein JavaScript mit dem Namen `script.js` im selben Verzeichnis wie das HTML-Dokument befindet. Das Skript wird hier gewöhnlich sofort ausgeführt, bevor der Webbrowser mit der Webseite fortfährt. Bei solchen externen Skripten können Sie den Ausführungszeitpunkt mit den Attributen `async` und `defer` beeinflussen. Beide Attribute werden in Tabelle 3.6 genauer beschrieben.

Zwar ist es noch etwas zu früh für Details in JavaScript, aber trotzdem sollte hier erwähnt werden, dass ein Skriptcode im `head`-Bereich einer HTML-Seite die Ladezeit verlängern kann, weil der Rest der Seite so lange blockiert wird, bis das JavaScript ausgeführt wurde. Daher ist es in der Praxis meistens sinnvoller, den Skriptcode am Ende der HTML-Datei, am günstigsten vor dem schließenden `<body>`-Tag, zu verwenden.

### 3.7.1 Die HTML-Attribute für das HTML-Element `<script>`

| Attribut             | Bedeutung   |
|----------------------|---|
| <code>async</code>   | Wenn Sie <code>async</code> verwenden, wird das Skript asynchron mit dem HTML-Dokument ausgeführt. Das Skript wird ausgeführt, während das HTML-Dokument geparsert wird. Dieses Attribut kann nur für externe Skripte verwendet werden. |
| <code>charset</code> | Legt die Zeichencodierung für das externe Skript fest.  |
| <code>defer</code>   | Verwenden Sie dieses Attribut, wird zuerst die Website geparsert und dann das Skript ausgeführt. Dieses Attribut kann nur für externe Skripte verwendet werden.   |
| <code>src</code>     | Damit geben Sie die URL zum externen Skript an.   |
| <code>type</code>    | Damit geben Sie den MIME-Typ für das Stylesheet an (hier meistens mit <code>text/javascript</code> oder <code>text/ecmascript</code> ).   |

**Tabelle 3.6** Die Attribute für das `<script>`-Element

## 3.8 Metainformationen für das Dokument mit `<meta>`

Mit dem `meta`-Element können Sie zusätzliche Angaben bzw. Daten über das HTML-Dokument im Kopfbereich zwischen `<head>` und `</head>` notieren. Dies können Anweisungen für den Webbrowser, den Webserver oder einen Webcrawler (auch Spider, Searchbot, Bot oder Suchrobot) sein. Auch wenn die Verwendung solcher `meta`-Elemente freiwillig ist, werden diese häufig angegeben. Es ist gerade für Einsteiger recht schwierig, die Übersicht über die vielen vorhandenen HTML-Attribute und die möglichen Attributwerte zu behalten, die Sie mit dem `meta`-Element verwenden können. Viele dieser Zusatzangaben sind gar nicht einheitlich standardisiert.

### Webcrawler

Ein Webcrawler ist eine Anwendung, die das Internet durchsucht und ganze Websites analysiert. Es sind verschiedene Arten von Webcrawlern unterwegs, die verschiedene Arten von Informationen sammeln. Auch Suchmaschinen verwenden einen Webcrawler, um Websites zu analysieren. Im Grunde ist das Prinzip recht ähnlich wie beim Internetsurfen, wo Sie über Hyperlinks von einer Webseite zu anderen URLs gelangen. Ein Webcrawler speichert diese URLs und besucht diese Seiten nach und nach. Die Websites werden mit einer Indexierung ausgewertet, um so eine Suche nach den entsprechenden Daten zu ermöglichen.

### 3.8.1 Die gebräuchlichsten Metaangaben

Ein `meta`-Element setzt sich in der Regel aus mindestens zwei Attributen zusammen. Entweder bestehen die Attribute aus einer `name/content`-Kombination oder aus einer `http-equiv/content`-Kombination. Zusätzlich gibt es eine spezielle Version für die Zeichencodierung.

#### »name/content«-Kombinationen – die frei definierbaren Metadaten

Das `meta`-Element, das das HTML-Attribut `name` enthält, kann im Grunde beliebige Informationen im HTML-Attribut `content` enthalten. Theoretisch könnten Sie selbst den Inhalt von `name` mit beliebigen Werten versehen. Trotzdem wurden in HTML einige Standardmetadaten für den Attributwert `name` festgelegt. Diese `name/content`-Kombinationen sind allerdings nicht für persönliche Informationen gedacht, sondern sollten lediglich Informationen zum HTML-Dokument enthalten. Ein einfaches Beispiel könnte wie folgt aussehen:

```
...
<head>
  <title>Frei definierbare Metadaten</title>
  <meta name="author" content="Max Mustermann">
  <meta name="keywords" content="Metadaten, meta, html">
  <meta charset="UTF-8">
</head>
...
```

Hier sehen Sie zwei typische name/content-Kombinationen. Das erste Beispiel definiert den Autor der Webseite und das zweite Paar Schlüsselwörter (keywords) für die Suchmaschinen. Hier könnten Sie noch beliebig viele weitere solcher meta-Elemente verwenden.

### »http-equiv/content«-Kombinationen – HTTP-Äquivalente

Die Angaben mit http-equiv (auch *Pragma-Direktive* genannt) waren für den Webserver zur Kommunikation gedacht. Der Webserver sollte diese Informationen auslesen und dann (ver einfacht) bei der Antwort an den Client (Webbrowser) die ausgelesene Information berücksichtigen und mit im HTTP-Response-Header verwenden. Allerdings parsen Webserver eigentlich keine HTML-Dokumente, daher liegt es wieder am Browser, wie diese Angaben verarbeitet werden. Hierzu ein einfaches Beispiel:

```
<!doctype html>
<html lang="de">
  <head>
    <title>HTTP-Äquivalente</title>
    <meta http-equiv="refresh" content="5">
    <meta charset="UTF-8">
  </head>
  <body>
    <p>Seite wird alle 5 Sekunden erneuert.</p>
  </body>
</html>
```

**Listing 3.7** /Beispiele/Kapitel003/3\_8\_1/index.html

Mit dem Wert refresh für das Attribut http-equiv und dem Wert 5 beim Attribut content sorgen Sie dafür, dass der Webbrowser veranlasst wird, die Webseite alle 5 Sekunden zu aktualisieren.

### Zeichencodierung für das HTML-Dokument festlegen

Neben dem name/content- und dem http-equiv/content-Paar gibt es eine dritte Möglichkeit, mit der Sie die Zeichencodierung (einfacher) festlegen können. Generell sollten Sie diese Angaben verwenden, wenn Sie eine Webseite erstellen, die in einer anderen Sprache als Englisch verfasst wurde. Die Rede ist von der Zeile mit dem meta-Element, das Sie in jedem Beispiel des Buches verwenden:

```
<meta charset="UTF-8">
```

Damit stellen Sie sicher, dass auch deutsche Umlaute und manch andere Sonderzeichen, dank dem Zeichensatzstandard UTF-8, richtig dargestellt werden. Neben dem Internet verwenden auch moderne Betriebssysteme UTF-8, und sofern Sie keinen Grund haben, einen anderen Zeichensatz zu verwenden, sollten Sie immer mit UTF-8 arbeiten.

### 3.8.2 Den Viewport einstellen

An dieser Stelle will ich auch gleich auf den Viewport voregreifen, da eine korrekte Einstellung verhindert, dass eine responsive Webseite auf dem mobilen Gerät in einer kleinen Ansicht dargestellt wird. Der Viewport ist der Bereich des Browserfensters, in dem die Webinhalte angezeigt werden. Ohne spezielle Vorkehrungen würden Webseiten auf den mobilen Browsern eines Smartphones so weit verkleinert, bis diese komplett auf den Bildschirm passen. Dadurch behalten die Besucherinnen und Besucher den Überblick und können in die Seite hineinzoomen.

Wenn Sie heute moderne Webseiten erstellen wollen, dann gehört die Berücksichtigung der verschiedenen Gerätegrößen und ein responsives Webdesign dazu. Bei der Erstellung responsiver Webseiten müssen Sie diese automatische Verkleinerung verhindern. Dies können Sie mit einem meta-Element wie folgt machen:

```
<meta name="viewport" content="width=device-width">
```

Damit weisen Sie den Browser an, nicht eine imaginäre Breite zu verwenden, sondern die tatsächliche Breite des Geräts. Das Ergebnis dieser Zeile bei einer responsiven Webseite können Sie in Abbildung 3.7 sehen, in der auf der linken Seite die automatische Verkleinerung durchgeführt und auf der rechten Seite der Viewport mit dem meta-Tag verwendet wurde.



**Abbildung 3.7** Eine responsive Webseite, links ohne und rechts mit dem Meta-Viewport

Auf den Viewport und das responsive Webdesign gehe ich in diesem Buch in Kapitel 13, »Responsive Layouts mit CSS erstellen«, noch gesondert ein. Ohne es an dieser Stelle genauer zu erklären, hat sich mittlerweile folgendes meta-Element dafür durchgesetzt:

```
<meta name="viewport"
      content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
```

Mit initial-scale=1.0 sorgen Sie dafür, dass der Browser die Seite mit der normalen Zoomstufe darstellt, und mit shrink-to-fit=no geben Sie dem Safaribrowser auf dem iPad die Anweisung, auch beim Split View nicht zu verkleinern.

### 3.8.3 Nützliche Metadaten für einen Webcrawler angeben

In diesem Abschnitt beschreibe ich kurz einige Metadaten für die Suchroboter von Suchmaschinen (Webcrawlers). Allerdings muss Ihnen hierbei klar sein, dass diese Angaben lediglich Empfehlungen für die Webcrawlers sind. Ob sich die Suchbots daran halten, liegt nicht in Ihrer Hand. Zumindest wurden diese Attributwerte teilweise von Google, Yahoo und Microsoft (mit)entworfen, und daher dürften sich diese Anbieter zumindest wohl daran halten. Wenn Sie Angaben für den Webcrawler als Metadaten aufnehmen wollen, müssen Sie dem Attribut name den Wert robots zuweisen. Im Attribut content schreiben Sie (oder schlagen vor), was der Webcrawler zu tun hat, wenn er die Webseite besucht, z. B.:

```
<meta name="robots" content="index,follow">
```

Damit erlauben Sie dem Suchrobot, die Webseite in den Suchmaschinenindex (index) aufzunehmen und den Hyperlinks auf der Seite zu folgen (follow). Allerdings können Sie sich diese Angaben gewöhnlich sparen, weil dies dem üblichen Verhalten eines Webcrawlers entspricht.

Falls Sie nicht wollen, dass die Seite indiziert oder dass den Hyperlinks gefolgt wird, können Sie in content die Attributwerte noindex und/odernofollow verwenden, z. B.:

```
<meta name="robots" content="noindex">
```

Hiermit geben Sie an, dass Ihre Webseite nicht in den Suchmaschinenindex aufgenommen werden soll (noindex), sodass die Seite nicht über eine Suchmaschine gefunden wird. Wollen Sie die Seite in den Suchmaschinenindex aufnehmen lassen, aber nicht, dass den Hyperlinks gefolgt wird, brauchen Sie nur den Attributwertnofollow in content zu verwenden.

### 3.8.4 Hilfreiche Metadaten für Suchmaschinen

Speziell für Suchmaschinen sind zwei name-Werte von Bedeutung, nämlich keywords und description. Allerdings hat der Wert keywords an Bedeutung verloren, weil er in der Vergangenheit missbraucht wurde, um Suchmaschinen mit vielen irreführenden Schlagwörtern zu füttern (*Keyword-Stuffing*), um so in der Suche möglichst hoch aufgelistet zu werden. Mittler-

weile indizieren die Suchmaschinen wieder gezielter den Inhalt einer Website und lassen die Schlüsselwörter eher unbeachtet (bzw. weniger beachtet) stehen. Wenn Sie trotzdem Schlüsselwörter angeben wollen, müssen Sie die einzelnen Schlagwörter in content mit Komma trennen, wie das folgende Beispiel zeigt:

```
<meta name="keywords" content="html, meta, keywords">
```

Hier wurden z. B. die Schlüsselwörter html, meta und keywords als Stichwörter für die Webseite verwendet.

Interessanter dagegen ist der Beschreibungstext der Webseite. Zwar wird dieser Text wohl eher nicht direkt bei den Suchergebnissen berücksichtigt, aber diese Beschreibung ist neben dem Titel das Erste, was der Anwender in der Suchmaschine als Information von Ihrer Webseite aufgelistet bekommt. Die Beschreibung sollten Sie möglichst kurz und präzise halten und maximal 150 bis 250 Zeichen verwenden (abhängig von der Suchmaschine). Ein zu langer Text wird gekürzt.

Hierzu ein Beispiel einer solchen Beschreibung:

```
...
<head>
  <title>Beschreibungstext für Suchmaschinen</title>
  <meta charset="UTF-8">
  <meta name="description"
        content="Eine Beschreibung sollte möglichst
        kurz und präzise sein. Hier sollte
        in 2-3 Sätzen zusammengefasst werden,
        worum es auf dieser Seite geht. Zu
        viele Zeichen werden gekürzt.">
</head>
<body>
  ...
</body>
...

```

Bei Google z. B. wird dieser Beschreibungstext in der Regel wie in Abbildung 3.8 aufgelistet:

#### Beschreibungstext für Suchmaschinen

www.woafu.de

Eine Beschreibung sollte möglichst kurz und präzise sein. Hier sollte in 2-3 Sätzen zusammengefasst werden, worum es auf dieser Seite geht. Zu viele Zeichen werden gekürzt. ...

**Abbildung 3.8** Neben dem <title>-Element gehört der Beschreibungstext häufig zu den ersten Merkmalen, die bei einer Suchmaschine angezeigt werden.

Wenn Sie keine Beschreibung mit einem `meta`-Element angeben, wird dieser Text aus den Teilen des Seiteninhalts generiert. Hierbei lässt es sich allerdings nicht genau vorhersagen, wie diese Beschreibung aussieht und was für ein Text dafür verwendet wird. Daher sollten Sie auf jeden Fall die Beschreibung selbst in die Hand nehmen und dies nicht dem Algorithmus einer Suchmaschine überlassen.

#### Der erste Eindruck ist wichtig

Auch wenn es nicht mehr so wichtig ist wie zu Beginn des Internets, spielen die Metadaten nach wie vor eine bedeutende Rolle bei der Erfassung der Suchmaschinen. Ihr Augenmerk sollten Sie daher stets auf das `title`-Element und die Beschreibung (`name="description"`) legen, weil diese Elemente häufig das Erste sind, was die Besucherinnen und Besucher der Webseite von den Suchmaschinen zurückbekommen, wenn die Seite bei einer Suche aufgelistet wird.

#### 3.8.5 Nützliche Metadaten für den Webbrowser

Wollen Sie den Inhalt einer Webseite nach einer bestimmten Zeit aktualisieren oder auf eine andere URL weiterleiten, können Sie hierfür das Attribut `http-equiv` mit dem Wert `refresh` verwenden. Über das `content`-Attribut können Sie die Zeit setzen, bis wann die Aktualisierung bzw. Weiterleitung erfolgen soll.

Eine Aktualisierung der Webseite erzwingen Sie wie folgt:

```
<meta http-equiv="refresh" content="30">
```

Damit würde die aktuell geladene Webseite alle 30 Sekunden aktualisiert.

Die Weiterleitung auf eine andere Website lässt sich recht ähnlich einrichten:

```
<meta http-equiv="refresh"
      content="5; URL=http://domain.de/">
```

Hiermit verlassen Sie den Browser nach 5 Sekunden, zur URL `domain.de` zu wechseln. Sie könnten hier auch 0 Sekunden verwenden, aber so können Sie zumindest den Anwender im HTML-Dokumentkörper darauf hinweisen, warum und wohin er weitergeleitet wird.

#### Automatische Weiterleitung nicht mehr verwenden

Eine automatische Weiterleitung kann hilfreich sein, wenn sich die Adresse des Webprojekts geändert hat. Allerdings ignorieren einige Browser je nach Einstellung diese Weiterleitung. Beachten Sie zudem auch, dass die Suchmaschinen diese Weiterleitung ignorieren. Hier ist es oftmals besser, einen Hyperlink mit Informationen im HTML-Dokumentkörper zur neuen URL

mit einem erklärenden Hinweis zu definieren. Auch bei der Zeitangabe 0 könnte es für den Besucher der Seite schwierig werden, wenn er die Zurück-Schaltfläche des Browsers verwendet, weil dieser somit immer wieder nach vorn geworfen würde. Alternativ lässt sich eine Weiterleitung auch auf dem Server erstellen. Wer z. B. Zugriff auf die Konfigurationsdatei `.htaccess` (beim Apache-Webserver) oder `web.config` (bei IIS) hat, kann dort eine Weiterleitung konfigurieren. Die automatische Weiterleitung wurde durch das W3C mittlerweile ohnehin als »deprecated« eingestuft, weshalb man für künftige Webprojekte auch Abstand davon nehmen sollte. Da Weiterleitungen immer noch verbreitet sind, habe ich das Thema hier trotzdem noch aufgenommen.

Wie bereits erwähnt, können Sie auch die alte Angabe zur Zeichencodierung verwenden:

```
<meta http-equiv="content-type"
      content="text/html; charset=utf-8" />
```

Diese Angabe entspricht der aktuelleren in HTML eingeführten Angabe von:

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

Die zusätzliche Verwendung der alten Angabe hat den Vorteil, dass diese auch von älteren Browsern verstanden wird, die `<meta charset="UTF-8" />` nicht kennen.

#### 3.8.6 Allgemeine Metadaten verwenden

Zusätzlich gibt es eine beträchtliche Anzahl von allgemeinen Metadaten, wie z. B. die Autorin oder den Autor des HTML-Dokuments oder wann das Dokument bearbeitet wurde. Dies ist z. B. hilfreich, wenn mehrere Personen an einem HTML-Projekt arbeiten. Diese Informationen können Sie alle als `name/content`-Kombination angeben. Hierzu einige Beispiele:

```
<meta name="author" content="Max Mustermann">
<meta name="date" content="2021-01-15T12:00:00+01:00">
```

Hier wurden der Verfasser der Webseite (`author`) und das Datum der letzten Änderung (`date`) angegeben. Wenn Sie persönliche Angaben für die Leserinnen und Leser zum aktuellen HTML-Dokument machen wollen, sollten Sie dies nicht mit den Metadaten machen, sondern direkt im HTML-Dokument in einer lesbaren Form. Diese Metadaten sind nur sinnvoll, wenn jemand den Quelltext des Dokuments betrachtet oder eine Software diese Daten ausliest. Auch hierzu gibt es noch weitere allgemeine Metaangaben wie `generator` für die Angabe, mit welcher Software die Webseite erstellt wurde. Oder machen Sie mit `application-name` spezielle Angaben, wenn die Webseite zu einer bestimmten Webplattform gehört oder wenn eine bestimmte Webanwendung in der Webseite ausgeführt wird.

**Für weitere Recherchen im Internet**

Eine Übersicht über die Standardmetaangaben finden Sie auf der Webseite: [www.whatwg.org/specs/web-apps/current-work/multipage/semantics.html#standard-metadata-names](http://www.whatwg.org/specs/web-apps/current-work/multipage/semantics.html#standard-metadata-names). Vorgeschlagene oder gegebenenfalls künftige Metaangaben finden Sie auf dieser Webseite: <http://wiki.whatwg.org/wiki/MetaExtensions>.

**3.8.7 Meine Empfehlung: Diese Metadaten gehören in das HTML-Grundgerüst**

Sie haben nun viele Metadaten kennengelernt und werden sich nun fragen, welche davon für die eigene Website in der Regel sinnvoll sind. Das bleibt Ihnen letztlich selbst überlassen, aber persönlich verwende ich für ein fertiges Projekt immer mindestens die Zeichencodierung für UTF-8, eine Seitenbeschreibung und den Viewport im head-Element:

```
...
<head>
    <title>Deutsche Umlaute</title>
...
    <meta charset="UTF-8" />
    <meta name="description" content="Eine Beschreibung sollte möglichst
        kurz und präzise sein. Hier sollte
        in 2-3 Sätzen zusammengefasst werden,
        worum es auf dieser Seite geht. Zu
        viele Zeichen werden gekürzt." />
    <meta name="viewport"
        content="width=device-width,initial-scale=1.0, shrink-to-fit=no" />
</head>
...
```

**Beispiele für das Buch bleiben reduziert**

Bei den Beispielen für das Buch habe ich meistens nur die Zeichencodierung verwendet, damit der Quelltext übersichtlicher bleibt. Der Viewport ist sinnvoll, wenn Sie eine responsive Website erstellen, was man in der Regel ja auch will. Die Seitenbeschreibung ist wichtig, wenn Sie die Website veröffentlichen und Suchmaschinen den Text als kurze Inhaltsangabe verwenden sollen.

**3.8.8 Die HTML-Attribute für das HTML-Element <meta>**

In der folgenden Tabelle finden Sie einen Überblick zu den HTML-Attributen für das meta-Element.

| Attribut   | Bedeutung   |
|------------|---|
| content    | Übergibt den Wert, der mit dem Attribut von http-equiv oder name verbunden ist.   |
| charset    | Legt die Zeichencodierung für das HTML-Dokument fest.   |
| http-equiv | Wird für den HTTP-Response-Header verwendet. Beispielsweise können Sie hiermit eine Webseite nach einer gewissen Zeit aktualisieren oder auf eine andere URL umleiten. Einige mögliche Werte sind: <ul style="list-style-type: none"> <li>▶ content-language</li> <li>▶ content-type</li> <li>▶ default-style</li> <li>▶ refresh</li> </ul> |
| name       | Definiert einen Namen für die Metadaten. Einige Standardwerte sind: <ul style="list-style-type: none"> <li>▶ application-name</li> <li>▶ author</li> <li>▶ description</li> <li>▶ generator</li> <li>▶ keywords</li> </ul>  |

Tabelle 3.7 Die Attribute für das <meta>-Element

**3.9 Zusammenfassung**

In diesem Kapitel haben Sie alle Elemente kennengelernt, die Sie im Kopf des HTML-Dokuments zwischen <head> und </head> notieren können. Um mit dem nächsten Kapitel fortfahren, ist es nicht unbedingt nötig, alle diese Elemente im Kopf zu haben. Zu einigen HTML-Elementen können Sie bei Bedarf jederzeit wieder zurückblättern. Die wichtigsten beiden Elemente in diesem Kapitel sind:

- ▶ Das title-Element: Allein schon für ein valides HTML benötigen Sie das title-Element im Kopf zwischen <head> und </head> des HTML-Dokuments. Das title-Element wird in der Kopfleiste des Webbrowsers oder in den Registern angezeigt. Außerdem wird es beim Setzen eines Lesezeichens als Namensvorschlag verwendet und auch von den Suchmaschinen als anklickbarer Verweis aufgelistet. Also genügend Argumente, das title-Element zu verwenden. Der Vollständigkeit halber empfiehlt es sich dann auch, die Seitenbeschreibung mit <meta name="description"> zu verwenden.
- ▶ Zeichencodierung mit <meta charset="UTF-8">: Mit dem meta-Element können Sie weitere Angaben oder Daten über das HTML-Dokument hinterlegen, etwa Anweisungen für den Webbrowser, Webserver oder die Suchmaschinen. Zwar können Sie hier eine Menge ver-