
Isolde Kommer

1. Ausgabe, März 2024

ISBN 978-3-98569-176-0

HTML5, CSS3 und JavaScript

Webseiten entwickeln –
Fortgeschrittene Anwendungen
(Stand 2024)

HTML5F_2024

Impressum

Matchcode: HTML5_2024

Autorin: Isolde Kommer

1. Ausgabe, März 2024

HERDT-Verlag für Bildungsmedien GmbH
Uwe-Zeidler-Ring 12
55294 Bodenheim
Internet: www.herd.com
E-Mail: info@herd.com

© HERDT-Verlag für Bildungsmedien GmbH, Bodenheim

Druck und Bindearbeiten: Esser printSolutions GmbH, D-75015 Bretten
Edubook AG, CH-5634 Merenschwand

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Wenn nicht explizit an anderer Stelle des Werkes aufgeführt, liegen die Copyrights an allen Screenshots beim HERDT-Verlag. Sollte es trotz intensiver Recherche nicht gelungen sein, alle weiteren Rechteinhaber der verwendeten Quellen und Abbildungen zu finden, bitten wir um kurze Nachricht an die Redaktion.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Die in diesem Buch und in den abgebildeten bzw. zum Download angebotenen Dateien genannten Personen und Organisationen, Adress- und Telekommunikationsangaben, Bankverbindungen etc. sind frei erfunden. Eventuelle Übereinstimmungen oder Ähnlichkeiten sind unbeabsichtigt und rein zufällig.

Die Bildungsmedien des HERDT-Verlags enthalten Verweise auf Webseiten Dritter. Diese Webseiten unterliegen der Haftung der jeweiligen Betreiber, wir haben keinerlei Einfluss auf die Gestaltung und die Inhalte dieser Webseiten. Bei der Bucherstellung haben wir die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Wir werden bei Kenntnis von Rechtsverstößen jedoch umgehend die entsprechenden Internetadressen aus dem Buch entfernen.

Die in den Bildungsmedien des HERDT-Verlags vorhandenen Internetadressen, Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen waren zum Zeitpunkt der Erstellung der jeweiligen Produkte aktuell und gültig. Sollten Sie die Webseiten nicht mehr unter den angegebenen Adressen finden, sind diese eventuell inzwischen komplett aus dem Internet genommen worden oder unter einer neuen Adresse zu finden. Sollten im vorliegenden Produkt vorhandene Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen nicht mehr der beschriebenen Software entsprechen, hat der Hersteller der jeweiligen Software nach Drucklegung Änderungen vorgenommen oder vorhandene Funktionen geändert oder entfernt.

Bevor Sie beginnen ...	4	7 Objektmodell in JavaScript	92
1 Navigation des Webauftritts	5	7.1 Einführung in das Objektmodell	92
1.1 Die Navigation planen	5	7.2 Objekt <code>navigator</code>	93
1.2 Navigationsleisten mit HTML und CSS erstellen	7	7.3 Objekt <code>window</code>	94
1.3 Navigationsmenüs erstellen	12	7.4 Objekt <code>document</code>	95
1.4 Imagemaps	18	7.5 Objekt <code>event</code>	97
1.5 Logische Verlinkung	20	7.6 Objekt <code>history</code>	99
1.6 Auf eine andere Adresse umleiten	22	7.7 Objekt <code>location</code>	100
1.7 Übungen	23	7.8 Vordefinierte Objekte in JavaScript	101
2 Bereiche definieren und positionieren	24	7.9 Übung	111
2.1 Zeitgemäßes Webseitenlayout	24	8 Zugriff auf HTML-Elemente	112
2.2 Container erstellen	25	8.1 Grundlagen zum <code>document</code> -Objekt	112
2.3 <code>div</code> -Container und semantische Elemente mit CSS formatieren	27	8.2 Unterobjekt <code>forms</code>	112
2.4 <code>div</code> -Container schweben lassen	29	8.3 Eingabefelder	114
2.5 <code>div</code> -Container positionieren	31	8.4 Kontroll- und Optionsfelder	115
2.6 Das Seitenlayout mit positionierten Elementen aufbauen	34	8.5 Auswahllisten	116
2.7 Übungen	39	8.6 Schaltflächen	118
3 Elemente mit CSS anordnen	41	8.7 Eingaben überprüfen	119
3.1 Weitergehende Möglichkeiten zur Anordnung von Elementen	41	8.8 Unterobjekt <code>images</code>	121
3.2 Ebenen anlegen	44	8.9 Übungen	125
3.3 Responsives Webdesign	46	9 DOM-Scripting	127
3.4 Übungen	52	9.1 Bestandteile	127
4 Sound und Videos einbinden	53	9.2 DOM-Standard ermitteln	127
4.1 Multimedia im Web	53	9.3 Container erstellen	128
4.2 Multimedia-Formate kennenlernen	55	9.4 Ebenen anzeigen und verbergen	129
4.3 Sound in HTML5 einbinden	56	9.5 Die Position von Ebenen zur Laufzeit ändern	131
4.4 Videos einbinden	58	9.6 Ebenen bewegen	133
4.5 Übungen	60	9.7 Umfangreiches Beispiel	1367
5 JavaScript	61	9.8 Übungen	143
5.1 Grundlegendes zu JavaScript	61	10 JavaScript-APIs der HTML5-Spezifikation	145
5.2 JavaScript in Webseiten einfügen	62	10.1 Was ist eine API?	145
5.3 Variablen und Datentypen	64	10.2 Häufig verwendete JavaScript-APIs	145
5.4 Operatoren	68	10.3 Die Geolocation-API	146
5.5 Kontrollstrukturen	72	10.4 Der Canvas-2D-Kontext	150
5.6 Kommentare	77	10.5 Text auf den Canvas zeichnen	155
5.7 Übungen	78	10.6 Übungen	158
6 Funktionen und Interaktionen in JavaScript	79	11 Kurzeinstieg in AJAX	159
6.1 Funktionen	79	11.1 Was steckt hinter AJAX?	159
6.2 Objektunabhängige Funktionen	86	11.2 Das <code>XMLHttpRequest</code> -Objekt	160
6.3 Interaktionen	88	11.3 Was ist noch möglich?	166
		Stichwortverzeichnis	168

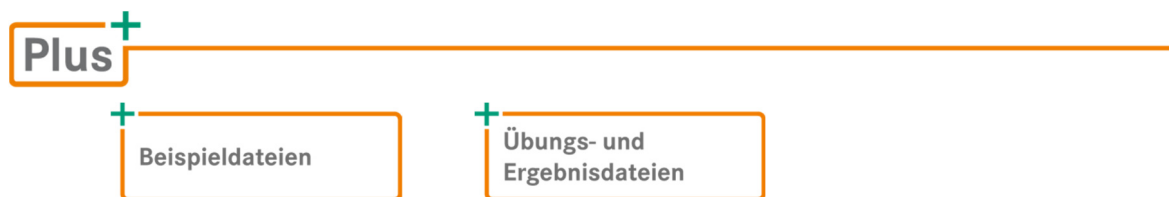
Bevor Sie beginnen ...

HERDT BuchPlus – unser Konzept:

Problemlos einsteigen – Effizient lernen – Zielgerichtet nachschlagen

(weitere Infos unter www.herdt.com/BuchPlus)

Nutzen Sie dabei unsere maßgeschneiderten, im Internet frei verfügbaren Medien:



Wie Sie schnell auf diese BuchPlus-Medien zugreifen können, erfahren Sie unter www.herdt.com/BuchPlus.

Was benötigen Sie zum Arbeiten mit diesem Buch?

Um die Arbeitsschritte in diesem Buch durchzuführen, benötigen Sie

- ✓ einen HTML5-fähigen, standardkonformen Webbrowser (z. B. eine aktuelle Version von Chrome, Firefox, Microsoft Edge, Safari oder Opera),
- ✓ einen Texteditor, wie zum Beispiel Windows Editor oder die Freeware Notepad++,
- ✓ für einige Beispiele einen aktiven Internetzugang und einen Testserver.

Für die Beispiele im Buch wurden folgende Browser verwendet:

- ✓ Google Chrome 122.0.6261.112
- ✓ Mozilla Firefox 123.0.1
- ✓ Microsoft Edge 122.0.2365.80

Das sollten Sie bereits können

Sie können sich den Kursinhalt am besten aneignen, wenn Sie bereits über folgende Kenntnisse verfügen:

- ✓ Sie besitzen gute Betriebssystem-Kenntnisse (z. B. Windows bzw. macOS).
- ✓ Sie sind sicher im Umgang mit einem Webbrowser.
- ✓ Sie besitzen HTML5- und CSS3-Grundkenntnisse. Wir empfehlen Ihnen dazu das HERDT-Buch *HTML5 – Grundlagen der Erstellung von Webseiten*.

Weitere empfehlenswerte HERDT-Bücher:

- ✓ *CSS – Cascading Style Sheets Level 3 – Grundlagen (CSS3_2024)*
- ✓ *JavaScript – Grundlagen (JAVS_2023)*

1

Navigation des Webauftritts

1.1 Die Navigation planen

Navigationselemente, Navigationsleisten, Navigationsmenüs

Ein Webauftritt, auch **Website** oder **Site** genannt, besteht aus mehreren miteinander verknüpften HTML-Seiten und eventuell Ressourcen wie PDF-Dokumenten oder downloadbaren Dateien, z. B. ZIP-Archiven.

Diese HTML-Seiten und Ressourcen werden mithilfe von **Navigationselementen**, z. B. Text-Hyperlinks und Buttons, miteinander vernetzt. Sobald der Nutzer auf ein Navigationselement klickt, steuert er die damit verknüpfte Webseite oder andere Ressource an.

Häufig werden die Navigationselemente übersichtlichen **Navigationsleisten** oder **Navigationsmenüs** zusammengefasst. Die meisten Webauftritte verfügen über eine oder mehrere solcher Navigationsleisten oder -menüs, mit deren Hilfe sich der Nutzer durch den Webauftritt bewegen kann.



Einstufige horizontale Navigationsleiste mit Text-Hyperlinks



Einstufige horizontale Navigationsleiste mit Buttons



Zweistufiges horizontales Navigationsmenü

Navigationselemente richtig platzieren

Die richtige Platzierung der Gesamtnavigation erleichtert dem Nutzer das Bewegen im Webauftritt. Im Lauf der Zeit haben sich bestimmte Konventionen für die Position der Navigationselemente herausgebildet. Diese Konventionen sollten Sie beachten, damit sich der Nutzer schnell und problemlos in Ihrem Webauftritt zurechtfindet.

Kaum ein Besucher ist bereit, beim Besuch eines Webauftritts zu überlegen, wo er sich gerade befindet und wohin er gelangen kann. Eine solche Website wird er daher bald verlassen.

- ① In der linken oberen Ecke befindet sich ein Link zur Startseite. Meist ist er in Form des Firmen- oder Webauftritt-Logos gestaltet.
- ② Die wichtigsten Sprungverweise werden in Form einer Navigationsleiste oder eines Navigationsmenüs horizontal entlang des oberen Randes erwartet. Möglich ist auch eine vertikale Anordnung entlang des linken Rands.
- ③ Bei stark untergliederten Webseiten wird die horizontale Navigation häufig als Hauptnavigation zu den übergeordneten Themen des Webauftritts verwendet. Die vertikale Navigation dient in diesem Fall als ergänzende Unternavigation, die zu den verschiedenen Seiten des aktuellen Unterthemas führt (siehe nebenstehende Abbildung).
- ④ Mit einer weiteren Navigationsleiste am unteren Seitenrand kann der Nutzer ergänzende Seiten mit FAQ, Kontaktmöglichkeiten, Impressum usw. ansteuern.



Typische Anordnung von Navigationselementen

Navigationselemente gestalten

✓ Symbole benennen und verwenden

Navigationselemente sollen möglichst sofort als solche erkennbar und allgemein verständlich sein. Versuchen Sie, eindeutige Hyperlinkbeschriftungen und/oder grafische Symbole zu finden, die möglichst jeder Nutzer versteht.

Ein typisches Beispiel ist das Briefsymbol, das im Web zum Standard für einen E-Mail-Hyperlink geworden ist.



✓ Einheitlichkeit

Damit der Nutzer schon nach kurzer Zeit problemlos durch Ihren Webauftritt navigieren kann, sollten sich die wichtigsten Navigationselemente stets an derselben Stelle der Seiten befinden. Darüber hinaus sollten sie auch auf jeder Seite gleich gestaltet sein, um eine intuitive Orientierung zu ermöglichen. Eine solche konsistente Navigation nennt man persistente oder globale Navigation.

✓ **Barrierefreiheit**

Bei der Planung der Navigationselemente sollten Sie auch berücksichtigen, auf welche Weise diese von den Suchmaschinen erfasst und von alternativen Ausgabegeräten – wie etwa Screenreadern für sehbehinderte Nutzer – dargestellt werden. Im Idealfall können Ihre Navigationselemente von allen Browsern, Suchmaschinen und alternativen Ausgabegeräten problemlos erfasst und ausgewertet werden. Man spricht dann von einer vollständigen Barrierefreiheit.



Mit JavaScript realisierte Navigationsmenüs sorgen eventuell dafür, dass nur die Startseite Ihres Webauftritts indiziert wird. Manche Ausgabegeräte haben Schwierigkeiten mit der Auswertung solcher Navigationsmenüs. Ähnliches gilt für Imagemaps. Wenn Sie sich für Navigationselemente mit JavaScript oder Java entscheiden, sollten Sie deshalb für alternative Navigationsmöglichkeiten sorgen, die von allen Ausgabegeräten angezeigt und von den Suchmaschinen erfasst werden können.

Barrierefreiheit durch Trennung von Struktur und Darstellung

Navigationsleisten und -menüs aus Text- oder grafischen Hyperlinks ohne JavaScript sind barrierefrei. Sie lernen, wie Sie ausschließlich in HTML und CSS erstellte Navigationsleisten und -menüs gestalten. Dabei verwenden Sie für die Strukturierung der Navigation reinen HTML-Code und für die attraktive Darstellung als Navigationsleiste oder -menü CSS-Formatierungen.

Diese strikte Trennung in der Vorgehensweise ist eine Vorbedingung für die problemlose Gestaltung barrierefreier Systeme: Entwickeln Sie zuerst die Inhalte und die Funktionalität bis zum Ende, testen Sie sie und beschäftigen sich danach mit der Gestaltung.

Ein weiterer Vorteil dieser Lösungen ist, dass sie schnell und ohne großen Aufwand anpassbar sind, besonders wenn Sie externe Stylesheets verwenden. Wenn sich das Site-Design ändern soll, müssen Sie nur einige CSS-Definitionen ändern, um für ein neues Erscheinungsbild Ihrer Navigation zu sorgen, diese beispielsweise statt vertikal nun horizontal anzuzeigen.

1.2 Navigationsleisten mit HTML und CSS erstellen

Horizontale Navigationsleiste aus nicht formatierten Text-Hyperlinks erstellen

Die einfachste Möglichkeit, eine Navigationsleiste zu erstellen, ist die Aneinanderreihung von Text- oder grafischen Hyperlinks.

Beispiel: Text-Navigationsleiste ohne Formatierung (*kap01\textnavigation.html*)

Der folgende Code und die zugehörige Abbildung zeigen eine horizontale Navigationsleiste aus Text-Hyperlinks (`<a> ... `):

```
<p><a href="home.html">Home</a> |
<a href="profil.html">Profil</a> |
<a href="service.html">Service</a> |
<a href="jobs.html">Jobs</a> |
<a href="impress.html">Impressum</a> |
<a href="hilfe.html">Hilfe</a></p>
```

[Home](#) [Profil](#) [Service](#) [Jobs](#) [Impressum](#) [Hilfe](#)

Horizontale Navigationsleiste mit unformatierten Text-Hyperlinks

Diese einfache Codestruktur genügt als Grundlage für ausgefeiltere Darstellungen, die in reinem CSS ohne JavaScript o. Ä. gestaltet werden können:

- ✓ Text-Navigationsleisten, die dynamisch auf die Mausektionen des Benutzers (Zeigen, Klicken) reagieren,
- ✓ unterschiedliche Button-Navigationsleisten, wahlweise auch mit Reaktion auf die Mausektionen des Benutzers.

Der Vorteil dieser Technik ist, dass solche Navigationselemente schnell geladen werden können und immer funktionieren – selbst in nicht CSS-fähigen oder Nur-Text-Browsern. Dort werden sie als Aneinanderreihung normaler Hyperlinks dargestellt.

In Kapitel 2 lernen Sie eine Möglichkeit kennen, wie Sie Ihren Navigationsbereich mithilfe des `<nav>`-Tags explizit innerhalb Ihrer Seite kennzeichnen.

Text-Navigationsleisten mit CSS gestalten

Um das Aussehen und die Reaktion der oben gezeigten Text-Navigationsleiste *text-navigation.html* zu verbessern, verwenden Sie die **Pseudoklassen** `a:link`, `a:visited`, `a:hover` und `a:active`. Mit diesen können Sie das Aussehen von Text-Hyperlinks während bestimmter und nach bestimmten Mausektionen des Anwenders ändern, also sogenannte Mouseover-Effekte erzeugen:

Pseudoklassen ermöglichen die CSS-Formatierung von HTML-Elementen, die sich nicht durch ein eindeutiges HTML-Tag ausdrücken lassen. Grundsätzlich geben Sie zuerst das jeweilige Element an, im Beispiel das Tag `<a>` für einen Hyperlink. Anschließend notieren Sie einen Doppelpunkt und eine definierte Angabe wie etwa `link` für einen noch nicht besuchten Hyperlink. Das Aussehen der Pseudoklassen legen Sie wie gewohnt mithilfe von CSS-Eigenschaften fest.

<code>a:link</code>	Der Hyperlink im ursprünglichen Zustand, bevor der Anwender ihn angeklickt oder mit der Maus darauf gezeigt hat
<code>a:visited</code>	Der Hyperlink auf eine bereits besuchte Seite
<code>a:hover</code>	Der Hyperlink beim Hovern (wenn der Benutzer mit der Maus darauf zeigt)
<code>a:active</code>	Der Hyperlink, den der Benutzer gerade anklickt

- Achten Sie darauf, bei der Definition der a-Pseudoklassen die in der Tabelle gezeigte Reihenfolge einzuhalten, damit sich die Angaben nicht gegenseitig außer Kraft setzen.

Beispiel: Text-Navigationsleiste mit Mouseover-Effekten (*kap01\mouseover.html*)

Im folgenden Beispiel versehen Sie die zuvor erzeugte einfache Text-Navigationsleiste mit Mouseover-Effekten, sodass sie optisch auf die Aktionen des Benutzers reagiert.

- ▶ Öffnen Sie die Beispieldatei *textnavigation.html*.
- ▶ Geben Sie direkt **vor** das schließende `</head>`-Tag den nachfolgend abgedruckten Code ein. Zu Übungszwecken legen Sie die CSS-Stildefinitionen damit in einem eingebetteten Stylesheet fest.

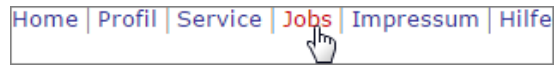
In der Praxis würden Sie statt eines eingebetteten eher ein externes Stylesheet erstellen und im Seitenkopf der einzelnen HTML-Seiten darauf verweisen. Dann könnten Sie bei Bedarf alle Navigationsleisten der Webseiten, in denen auf das externe Stylesheet verwiesen wird, gleichzeitig neu formatieren. Sie müssten dazu nur die Stildefinitionen in der externen Stylesheet-Datei abändern.

```
<style type="text/css">
<!--
① a:link,a:visited,a:hover,a:active {
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:small;
②    text-decoration:none;
}
③ a:link {
    color:#339;
}
a:visited {
    color:#909;
}
a:hover {
    color:#C00;
}
a:active {
    color:#999;
}
-->
</style>
```

- ① Zuerst definieren Sie die Eigenschaften, die allen Hyperlink-Zuständen gemeinsam sind: die Schriftfamilie (`font-family`) und die Schriftgröße (`font-size`). Dazu fassen Sie alle vier Pseudoklassen für die einzelnen Zustände des `<a>`-Tags zu einer einzigen Stildefinition zusammen. Die einzelnen Pseudoklassen trennen Sie dabei durch ein Komma.
- ② Mit `text-decoration:none` stellen Sie alle vier Zustände der Hyperlinks im Browser ohne Unterstreichung dar.

- ③ Anschließend notieren Sie die vier Pseudoklassen noch einmal gesondert und geben jeweils die Merkmale an, die sich bei den verschiedenen Hyperlink-Zuständen ändern sollen, im Beispiel die Textfarbe (`color`).

Die Auswirkungen von `visited` sehen Sie, wenn Sie auf den Link klicken und die Maustaste gedrückt halten.



Text-Navigationsleiste mit formatierten Hyperlinks

Button-Navigationsleisten mit CSS gestalten

Die einfache Text-Navigationsleiste im Beispiel *textnavigation.html* kann mit CSS so formatiert werden, dass die einzelnen Hyperlinks das Aussehen von dreidimensionalen Buttons annehmen. Auch dazu verwenden Sie die Pseudoklassen `a:link`, `a:visited`, `a:hover` und `a:active`.

Beispiel: Button-Navigationsleiste (*kap01\3d-navigation.html*)

Im folgenden Beispiel versehen Sie die verschiedenen Hyperlink-Zustände der einfachen Text-Navigationsleiste *textnavigation.html* mit unterschiedlichen Füllungen und Rahmenfarben, so dass sie wie grafische Buttons aussehen, die beim Anklicken gedrückt erscheinen und ansonsten hervortreten.

- ▶ Öffnen Sie die Beispieldatei *textnavigation.html*.
- ▶ Löschen Sie die senkrechten Striche (|) und alle Leerzeichen zwischen den einzelnen Hyperlinks.
- ▶ Geben Sie innerhalb des Tagpaars `<head> ... </head>` den nachfolgenden Code ein.

```

<style type="text/css">
<!--
① a:link, a:visited, a:hover, a:active {
    border-style:solid;
    border-width:thin;
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:small;
    padding:4px 8px;
    text-align:center;
    text-decoration:none;
    text-transform:uppercase;
}
② a:link {
    background-color:#BDB76B;
    border-color:#F0E68C olive olive #F0E68C;
    color:#000;
}
③ a:visited {
    background-color:#BDB76B;
    border-color:#F0E68C olive olive #F0E68C;
    color:#8B008B;
}

```

```

④ a:hover {
    background-color:#F0E68C;
    border-color:olive #F0E68C #F0E68C olive;
    color:#FF8C00;
}
⑤ a:active {
    background-color:#BDB76B;
    border-color:#F0E68C olive olive #F0E68C;
    color:#F0E68C;
}
-->
</style>

```

- ① Wie im vorigen Beispiel *mouseover.html* definieren Sie zuerst die Eigenschaften, die sämtlichen Hyperlink-Zuständen gemeinsam sind, beispielsweise die Schriftmerkmale, die Rahmenart (`solid` = durchgezogen) und die Rahmenbreite (`thin`). Für einen entsprechenden Abstand zwischen Text und Buttonkanten sorgt die Eigenschaft `padding`.
- ② Jetzt folgen die Eigenschaften für den Hyperlink im Normalzustand. Mit `background-color` bestimmen Sie die Hintergrundfarbe des Buttons, mit `color` die Textfarbe. `border-color` definiert die Rahmenfarbe: Der linke und der obere Rahmen werden khaki (`#F0E68C`), der rechte und der untere Rahmen olivfarben. Durch die Helligkeitsunterschiede entsteht der dreidimensionale Eindruck.
- ③ Nach demselben Muster formatieren Sie den besuchten Hyperlink. Ändern Sie nur die Textfarbe (`color`).
- ④ Der Link beim Hovern erhält eine deutlich andere Hintergrundfarbe. Ändern Sie auch die Textfarbe. Für die Rahmenfarbe (`border-color`) kehren Sie die Verhältnisse um: Der rechte und der untere Rahmen erhalten das hellere Khaki (`#F0E68C`), der linke und der obere Rahmen das dunklere Oliv.
- ⑤ Für den aktiven Link wählen Sie eine jeweils andere Hintergrund- und Textfarbe. Die Rahmenfarbe stellen Sie wieder ein wie beim Normalzustand und beim besuchten Hyperlink.



Button-Navigationsleiste in HTML und CSS

Button-Navigationsleisten mit CSS und Bilddateien gestalten

Bei Bedarf können Sie die zuvor gezeigte Technik zur Gestaltung von Button-Navigationsleisten mit CSS durch Bilder ergänzen. Dadurch ergeben sich weitere Gestaltungsmöglichkeiten, beispielsweise mit Farbverläufen oder Texturen.

Kann das Bild aus irgendeinem Grund nicht geladen werden, schadet dies der Funktionalität der Navigationsleiste nicht. Statt des Bildes wird dann einfach der hellere Hintergrund angezeigt.

Beispiel: Button-Navigationsleiste mit Hintergrundbild (*kap01\3d-navigation-bild.html, burst.jpg*)

Im folgenden Beispiel erhalten die als Buttons dargestellten Hyperlinks beim Hovern ein Hintergrundbild mit einem radialen Farbverlauf.

- ▶ Speichern Sie die Beispieldatei *3d-navigation.html* aus dem vorigen Beispiel und die Beispieldatei *burst.jpg* in demselben Ordner.
- ▶ Öffnen Sie die Datei *3d-navigation.html*.
- ▶ Ändern Sie die Definition der Pseudoklasse `a : hover` folgendermaßen ab:

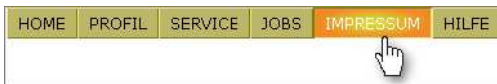


```

1 a: hover {
    background-color: #F0E68C;
    background-image: url (burst.jpg);
    border-color: olive #F0E68C #F0E68C olive;
    color: #FFFFFF;
}

```

- ① Die Pseudoklasse `a : hover` wurde durch die Eigenschaft `background-image` um das Bild `burst.jpg` ergänzt.



Button-Navigationsleiste mit Hintergrundbild beim Hovern

1.3 Navigationsmenüs erstellen

Nicht nur Navigationsleisten, sondern auch mehrstufige Navigationsmenüs lassen sich in HTML strukturieren und mit CSS gestalten.

Durch diese Möglichkeit können Sie völlig auf JavaScript und andere Technologien verzichten, die häufig zur Erstellung von Navigationsmenüs verwendet werden. Sie stellen damit sicher, dass Ihre Navigationsmenüs sowohl von Suchmaschinen als auch von den alternativen Ausgabegeräten und von Nur-Text-Browsern problemlos ausgewertet werden können.



Horizontales Navigationsmenü

Die HTML-Struktur eines Navigationsmenüs erstellen


Als Grundlage für das Navigationsmenü bietet sich eine unsortierte HTML-Liste an.

Beispiel: HTML-Menüstruktur (*kap01\menuestruktur.html*)

```

① <ul>
  <li><a href="#">Über uns</a>
  ② <ul>
    <li><a href="philos.html">Philosophie</a></li>
    <li><a href="firmeng.html">Firmengeschichte</a></li>
  </ul>
</li>
  ③ <li><a href="#">Produkte</a>
    <ul>
      <li><a href="buecher.html">Bücher</a></li>
      <li><a href="zeitschr.html">Zeitschriften</a></li>
      <li><a href="software.html">Software</a></li>
    </ul>
</li>
  <li><a href="#">Kontakt</a>
    <ul>
      <li><a href="beratung.html">Beratung</a></li>
      <li><a href="vertrieb.html">Vertrieb</a></li>
    </ul>
</li>
</ul>

```

- ① Geben Sie zunächst alle Hauptmenüpunkte als ungeordnete Liste ein.
- ② Diejenigen Punkte, die später im Untermenü erscheinen sollen, geben Sie als untergeordnete Liste ein.
- ③ Um eine vollständige Barrierefreiheit zu erzielen, sollten Sie auch die Hauptpunkte mit Hyperlinks versehen. Denn nur dann lässt sich das Navigationsmenü per -Taste navigieren. Das ist wichtig für eine korrekte Ausgabe auf alternativen Ausgabegeräten. Da die Hauptpunkte keine echten Sprungverweise enthalten, behelfen Sie sich mit Null-Hyperlinks (`href="#"`), Hyperlinks, die zwar als solche erkannt werden, aber nirgendwohin führen.

```

• Über uns
  • Philosophie
  • Firmengeschichte
• Produkte
  • Bücher
  • Zeitschriften
  • Software
• Kontakt
  • Beratung
  • Vertrieb

```

Fertige Menüstruktur

Von der inhaltlichen Seite her ist das Navigationsmenü fertig. Es funktioniert in allen Browsern und mit sämtlichen alternativen Ausgabegeräten. Sie können sich jetzt mit der Anordnung der Menüpunkte und danach mit der Gestaltung beschäftigen.

Das Navigationsmenü mit CSS anordnen

Für die Anordnung der Menüpunkte verwenden Sie CSS.

Beispiel: CSS-Menüanordnung (*kap01\menueanordnung.html*)

Zuerst entfernen Sie die Listenpunkte vor den einzelnen Elementen und definieren Schrift und Positionierung. Mit der Positionierung ordnen Sie die Liste in Menüform an. Erstellen Sie dazu im head-Bereich die folgende CSS-Regel:

```

<style type=text/css>
<!--
① ul {
②     font-family:Verdana, Arial, Helvetica, sans-serif;
③     font-size:x-small;
④     list-style:none;
⑤     margin:0;
⑥     padding:0;
⑦ }
⑧ li {
⑨     float:left;
⑩     position:relative;
⑪     width:10em;
⑫ }
⑬ li ul {
⑭     display:none;
⑮     position:absolute;
⑯     top:1.2em;
⑰     left:0;
⑱ }
⑲ li:hover ul {
⑳     display:block;
㉑ }
-->
</style>

```

- ① Notieren Sie die Definition des Tags `ul` (die unsortierte Liste).
- ② Geben Sie die gewünschten Schriftformatierungen an.
- ③ Als `list-style` wählen Sie `none`, um die Aufzählungspunkte vor den einzelnen Elementen zu entfernen.
- ④ `margin` und `padding` weisen Sie jeweils den Wert 0 zu. Damit bestimmen Sie, dass das Navigationsmenü am linken oberen Rand des Fensters beginnt.
- ⑤ Diese formatierte Liste verwandeln Sie nun in ein horizontales Menü. Dazu definieren Sie das Tag `li` neu, das die einzelnen Listenpunkte erzeugt.
- ⑥ Notieren Sie `float:left` mit einer Breite von 10 em und einer relativen Position. Nur dann werden die verschachtelten Elemente relativ zu den Hauptpunkten angeordnet.



Voll funktionsfähige Navigationsleiste

Ohne weitere Angaben werden HTML-Elemente im Browser in derselben Reihenfolge wie im Quelltext angezeigt. Mit der CSS-Eigenschaft `float` können Sie Elemente jedoch von Text umfließen lassen. Mit `float: left` steht das Element links und wird rechts umflossen. Detaillierte Informationen über `float` erhalten Sie in Kapitel 2.

Enthält Ihr Navigationsmenü längere Einträge, erhöhen Sie den Wert `10em` entsprechend.

- ⑦ Als Nächstes definieren Sie die Darstellung der Listenelemente der zweiten Ebene, also der zunächst unsichtbaren Menüelemente. Mit `li ul` erzeugen Sie eine Stildefinition, die nur für innerhalb eines `li`-Tags geschachtelte `ul`-Elemente gilt, also nur für die eingezogenen Unterpunkte.
- ⑧ Setzen Sie `display` auf `none`, damit die eingezogenen Listenelemente zunächst unsichtbar sind.
- ⑨ Positionieren Sie die Elemente anschließend absolut mit der Platzierung `1,2em` vom oberen und `0` von den übergeordneten Einträgen.
- ⑩ Mit der Pseudoklasse `li: hover` machen Sie das Navigationsmenü schließlich funktionsfähig. Mit `li: hover ul` sprechen Sie nur `ul`-Unterpunkte an, sobald sich die Maus über dem übergeordneten Hauptpunkt befindet. Mit `display: block` bestimmen Sie, dass die jeweiligen Unterpunkte beim Hovern über dem Hauptpunkt angezeigt werden.

Das Navigationsmenü mit CSS gestalten

Das Navigationsmenü ist bisher einfach gehalten. Sie können aber alle von CSS gebotenen Gestaltungsmöglichkeiten einsetzen, um das Navigationsmenü an das Design Ihres Webauftritts anzupassen.

Beispiel: CSS-Menügestaltung (*kap01\menuegestaltung.html*)

```
<style type=text/css>
<!--
ul {
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:small;
    list-style:none;
    margin:0;
    padding:0;
}
① li {
    background-color:#999;
    border-bottom-color:#F90;
    border-bottom-style:solid;
    border-bottom-width:.1em;
    border-right-color:#F90;
    border-right-style:solid;
    border-right-width:thick;
    float:left;
    padding:.5em;
    position:relative;
    width:13em;
}
}
```