

KAPITEL 1 AUF DIE PLÄTZE

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Titel der Webseite</title>
8      <link rel="icon" href="bilder/icon.ico">
9      <link rel="stylesheet" type="text/css" href="css/style.css">
10 </head>
11
12 <body>
13     <article>
14         <header>
15             </header>
16         <section>
17             Überraschung
18         </section>
19     </article>
20 </body>
21
22 </html>
```

SEI TEIL DES INTERNETS. Das Web ist eine Sammlung von untereinander vernetzten Dokumenten. In diesem Kapitel lernst du, wie du in wenigen Schritten ein solches Dokument erstellen kannst.

ERSTE SCHritte MIT HTML

Wofür steht HTML? HTML ist die Abkürzung für die Beschreibungssprache *HyperText Markup Language*. Sie dient dazu, die Struktur einer Webseite zu beschreiben. Also die Gliederung der Seite in Überschriften, Absätze und so weiter. Man kann zusätzlich in einem HTML-Dokument *Hyperlinks* (Verlinkungen) zu anderen Webseiten setzen oder multimediale Komponenten wie Text, Ton, Videos, Bilder oder Animationen einbinden.

BAUSTEINE EINES HTML-DOKUMENTS

Ein HTML-Dokument besteht aus einem `head` (Kopf) und einem `body` (Körper). Man kann diese Aufteilung mit diesem Buch vergleichen: Auf dem Buchumschlag findest du alles über das Buch, also beispielsweise den Autor und den Titel. Diese Informationen kommen in den `head`-Bereich. Wenn du das Buch aufschlägst, findest du den Inhalt des Buches mit Illustrationen, Texten oder Listen, und solche Inhalte können im HTML-Dokument im `body`-Bereich gefunden werden.

WAS IST EIN TAG?

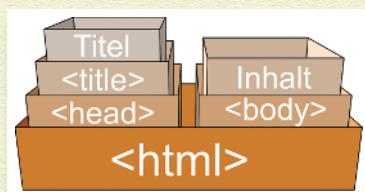
Bei `head` und `body` handelt es sich um *Tags*. Tags sind Markierungen, um den Anfang und das Ende eines Elementes zu kennzeichnen. Start- und End-Tags unterscheiden sich durch einen Schrägstrich.

In dem folgenden Codeabschnitt findest du die Start-Tags `<html>`, `<head>`, `<title>` und `<body>` und die End-Tags `</html>`, `</head>`, `</title>` und `</body>`.

```
<!DOCTYPE html>
<html>
<head>
    <title>Titel der Webseite</title>
</head>
<body>
    Inhalt der Webseite
</body>
</html>
```

TAGS VERSCHACHTELN

Die Tags in dem Codebeispiel sehen auf den ersten Blick etwas durcheinander aus, aber das liegt daran, dass die Tags ineinander verschachtelt sind.



Zwischen die Tags `<html>` und `</html>` kommen alle anderen Tags verschachtelt rein. Sie bilden also den äußereren Rahmen bzw. die größte Schachtel.



`<!DOCTYPE html>` setzt die Dokumentart auf HTML fest und gehört in jedes HTML-Dokument ganz oben in die allererste Zeile.

Der Tag `<head>` sorgt für die erste Verschachtelung. Innerhalb der head-Schachtel kann eine weitere Schachtel platziert werden, zum Beispiel `<title>` (Titel).

Zwischen `<title>` und seinem schließenden Tag `</title>` steht im Codebeispiel »Titel der Webseite«, diesen Text wird man auf einer Webseite im Tab des Browsers lesen können.



Danach wird der head-Tag geschlossen und der `<body>`-Bereich beginnt. Zwischen den body-Tags steht im Codebeispiel »Inhalt der Webseite«. Diesen Text findet man auf der Webseite selbst wieder.



Das HTML-Dokument endet mit einem schließenden body-Tag, bevor auch der html-Tag geschlossen wird.

DIE ERSTE WEBSEITE ERSTELLEN

- Um ein HTML-Dokument anzulegen, öffnest du den Texteditor deines Vertrauens.

Für den Anfang genügt irgendein beliebiger Texteditor auf deinem Computer, beispielsweise Editor (Windows) oderTextEdit (Mac).

2 Tippe das Codebeispiel von oben in dein Dokument ab.

3 Ergänze im head-Bereich die Zeile

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

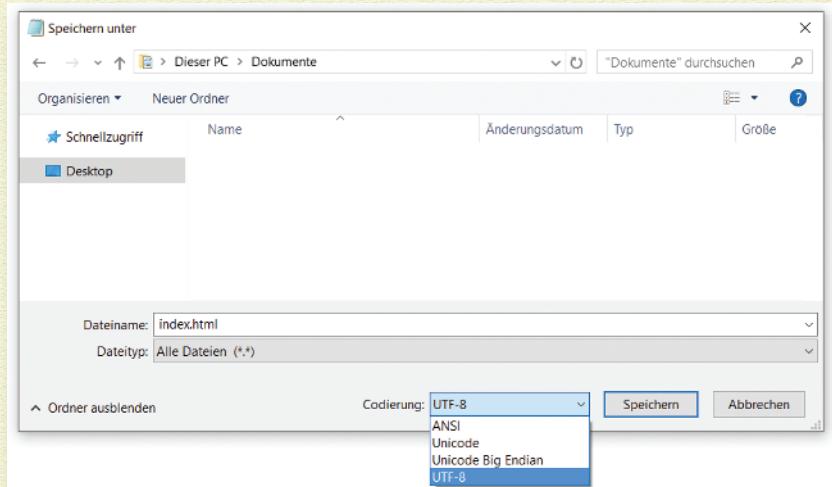


Ein Tag kann mit HTML-Attributen ergänzt werden, um zusätzliche Anweisungen zu geben. HTML-Attribute setzen sich aus einem Attributnamen und einem Attributwert in Anführungszeichen zusammen. Dazwischen steht ein Gleichheitszeichen:

`<tag attributname="wert"> ... </tag>`

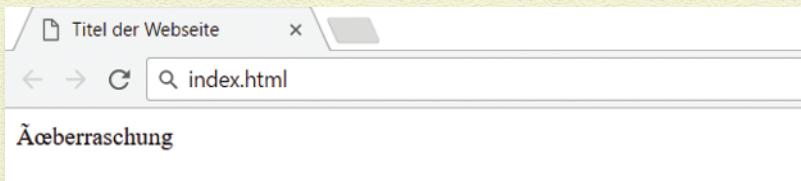
4 Speichere das Dokument unter `index.html` ab.

Achte darauf, dass die **Codierung auf utf-8 eingestellt ist.**



Du kannst das kontrollieren, indem du im HTML-Dokument »Inhalt der Webseite« durch »Überraschung« ersetzt. Falls du die falsche Codierung gewählt hast, erwartet dich auf

deiner Webseite im wahrsten Sinne des Wortes **eine kleine Überraschung.**



Das liegt daran, dass das Ü ohne entsprechende Zeichencodierung nicht richtig angezeigt werden kann.

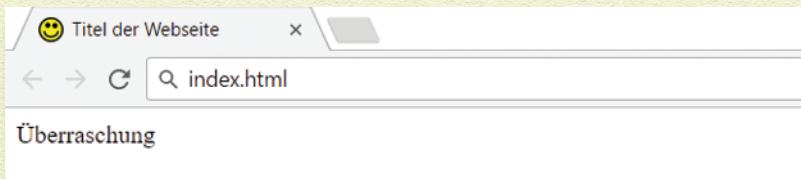
5 Öffne die index.html-Datei mit **deinem Browser.**



*Um Änderungen an deiner Webseite zu sehen, musst du sie zuerst wieder speichern und dann in **deinem Browser neu laden.***

EIN ICON FÜR DEN BROWSER-TAB

Neben dem Titel findest du auf Webseiten im **Browser-Tab** oft ein kleines Bildchen, ein sogenanntes *Icon*.



1 Lege einen Ordner **bilder** im selben Ordner an, in dem deine HTML-Datei liegt.

2 Speichere dort ein passendes Icon als **icon.ico** ab.



Unter www.wiley-vch.de/publish/dt/books/ISBN3-527-71403-0 findest du die Bilder und Icons, die wir für unsere Beispiele in diesem Buch benutzen.

3 Ergänze im head-Bereich folgende Zeile:

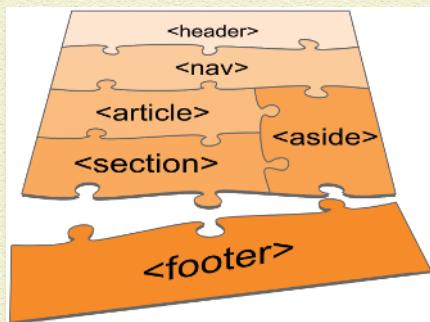
```
<link rel="icon" href="bilder/icon.ico" />
```



Wenn du kein Icon angezeigt bekommst, dann solltest du prüfen, ob du den Dateipfad, welcher nach href angegeben wird, richtig geschrieben hast. Hast du wirklich »bilder« getippt und heißt der Ordner auch so? Liegt der bilder-Ordner im selben Ordner wie die HTML-Datei? Hast du wirklich »icon.ico« getippt und heißt deine Bilddatei auch so?

BEREICHE IM <BODY>

Damit das HTML-Dokument übersichtlicher wird, gibt es verschiedene Tags, mit denen man den body-Bereich in weitere Teilbereiche unterteilen kann.



Der Name des Tags verrät, was du in diesem Bereich unterbringen solltest oder was du in dem Bereich vorfinden wirst. In der Tabelle findest du eine kleine Erläuterung dieser Tags.

Tag	Beschreibung
<article> (Artikel)	Inhalt des Dokuments
<aside> (Nebenbemerkung)	Zusatzinformation zum Inhalt der Webseite
<footer> (Fußzeile)	Kontakt-, Datenschutz- oder Entwicklerinformationen
<header> (Kopfzeile)	Überschriften, Datum, Uhrzeit, Logo und Autor
<nav> (Navigation)	Links zu anderen Webseiten oder Bereichen innerhalb der Seite
<section> (Abschnitt)	Gruppierungen im Inhalt

Nachdem du einige dieser Tags in dein Dokument eingebaut hast, könnte dein body-Bereich nun so aussehen:

```
<body>
  <article>
    <header>
    </header>
    <section>
      Überraschung
    </section>
  </article>
</body>
```

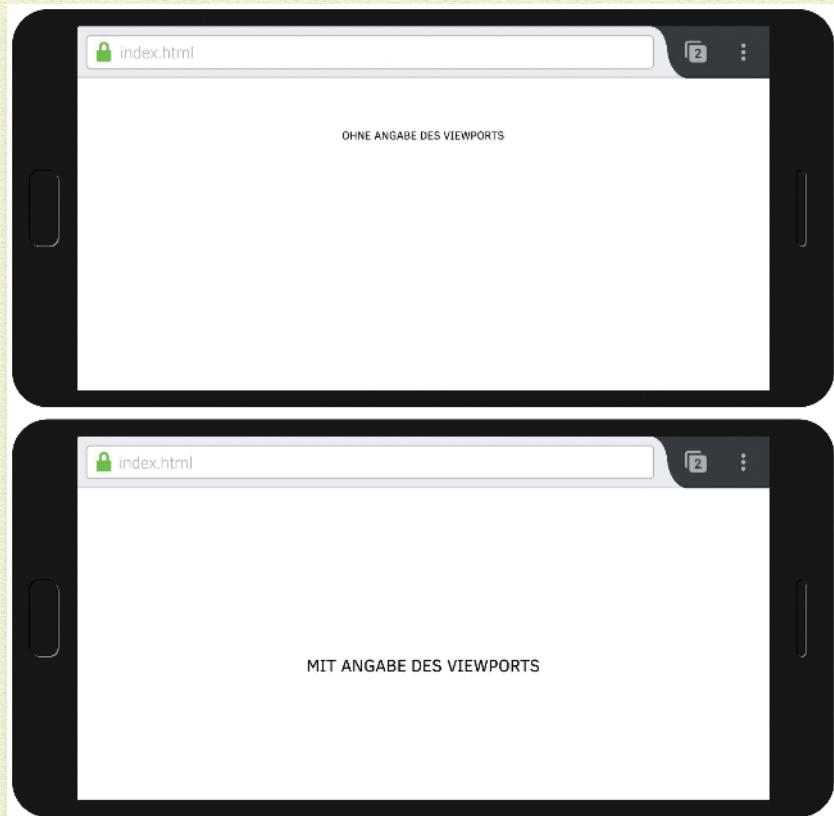
Am Aussehen deiner Webseite hat sich dadurch nichts geändert.

MOBILE GERÄTE UNTERSTÜTZEN

Mit *Viewport* (Ansichtsfenster) bezeichnet man den sichtbaren Bereich einer Webseite, den du als Besucher siehst. Auf einem Smartphone ist dieser Bereich deutlich kleiner als auf einem Laptop- oder Computerbildschirm.

Damit deine HTML-Seite an das jeweilige Gerät angepasst wird, setzt du im head-Bereich mit der folgenden Anweisung die Breite des Viewports gleich der Gerätebreite:

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0" />.
```



MIT CSS WIRD ES BUNT

Alle Elemente wurden bisher automatisch gestaltet, die Profis sagen »formatiert«: So ist die Schrift schwarz, der Text links angeordnet und der Hintergrund weiß. Aber woher kommen

diese Formatierungen? Ganz einfach: Jeder Browser besitzt eine eingebaute Gestaltungsanweisung, das sogenannte *Stylesheet* (Formatvorlage).

Und wenn man eine andere Schrift und eine andere Farbe haben möchte? Dafür gibt es CSS. CSS steht für *Cascading Style Sheets* und überschreibt das eingebaute Stylesheet des Browsers. Somit kannst du deine Seite individuell gestalten.

CSS-DOKUMENT EINBINDEN

- 1 Lege als Erstes im selben Ordner, in dem das HTML-Dokument liegt, einen Ordner namens `css` an.**
- 2 Erstelle im Editor eine neue leere Datei. Speichere die neue Datei im `css`-Ordner unter dem Namen `style.css`.**

Damit hast du ein CSS-Dokument erstellt.

- 3 Damit dein HTML-Dokument das CSS-Dokument auch benutzt, musst du im head-Bereich des HTML-Dokuments folgende Zeile ergänzen:**

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

HTML-ELEMENTE FORMATIEREN

In das CSS-Dokument schreibst du Befehle, die die Eigenschaften und damit das Aussehen bestimmter HTML-Elemente verändern.

Ein solcher Befehl besteht aus Selektor, Eigenschaft und Wert.
»select« ist das englische Wort für »auswählen«.)

```
Selektor{
  Eigenschaft1: Wert1;
  Eigenschaft2: Wert2;
  Eigenschaft3: Wert3;
}
```

Anstelle von *Selektor* schreibst du den Namen des Tags, auf den sich dein Befehl beziehen soll, beispielsweise »article«.

Dann suchst du dir eine Eigenschaft aus. Du kannst mit *box-shadow* (Box-Schatten) einen Schatten um das Element setzen.



Und dann musst du noch die Werte für die Eigenschaft setzen. Bei der Eigenschaft »box-shadow« sind drei Werte möglich: Die ersten beiden Werte stehen für die Verschiebung des Schattens nach rechts oder links und nach unten oder oben. Sind die beiden Werte 0, dann bleibt der Schatten direkt unter dem Element. Setzt du den ersten Wert hoch, also beispielsweise auf 10px, dann bewegt sich der Schatten nach rechts. Setzt du den zweiten Wert hoch, beispielsweise auf 15px, dann bewegt sich der Schatten nach unten. Der dritte Wert setzt den Übergang des Schattens. Je höher der dritte Wert ist, desto weicher sieht der Schatten aus.

Der vollständige Selektor sieht jetzt so aus:

```
article{  
    box-shadow: 0 0 20px;  
}
```



Du kannst für die ersten beiden Werte auch negative Zahlen angeben. Dann bewegt sich der Schatten nach links und nach oben.



Die Schattenfarbe ist automatisch Schwarz. Wenn du diese ändern willst, ergänzt du als vierten Wert eine Farbe. Dafür gibt es viele vordefinierte Farben, beispielsweise black (Schwarz), grey (Grau), blue (Blau), red (Rot), yellow (Gelb) oder white (Weiß).



Die ersten beiden Werte haben keine Einheit, denn wenn der Wert gleich 0 ist, brauchst du keine setzen. Der dritte Wert hat die Einheit px, welche für Pixel steht.

Eine weitere Eigenschaft, die du festlegen kannst, ist die Breite eines Elements: `width` (Breite). Den Wert kannst du wieder in Pixeln angeben. Mit `1100px` erzeugst du eine schmalere Box.



Der vollständige Selektor sieht jetzt so aus.

```
article{
  width: 1100px;
  box-shadow: 0 0 20px;
}
```

Und zu guter Letzt legst du noch den Außenabstand der Elemente fest, und zwar für den body und für den article.

Der Inhalt der Webseite liegt im body. Damit dafür die volle Breite und Höhe zur Verfügung steht, setzt du `margin` (Außenabstand) auf `0`. Beim article setzt du »margin« auf `auto`, Mit »auto« wird das Element waagerecht zentriert.



```
body{
  margin: 0;
}
article{
  margin: auto;
  width: 1100px;
  box-shadow: 0 0 20px;
}
```

DAS BOX-MODELL

Gerade hast du gelernt, dass man mit »margin« den Außenabstand eines HTML-Elements verändern kann. Das liegt daran, dass (fast) jedes HTML-Element eine Box ist.

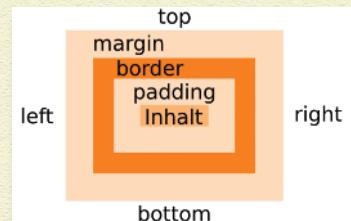
Es gibt also einen Außenabstand nach links (`margin-left`), nach rechts (`margin-right`), nach oben (`margin-top`) oder nach unten (`margin-bottom`).

Für alle diese Abstände kannst du verschiedene Werte festlegen.

Alternativ kannst du die Werte auch alle in eine Zeile schreiben. Dann musst du aber die Reihenfolge beachten, diese ist: top, right, bottom, left.

```
margin-top: 5px;  
margin-right: 10px;  
margin-bottom: 15px;  
margin-left: 20px;
```

```
margin: 20px 17px 0px 13px;
```



Die Box hat außerdem einen `border` (Rahmen) und einen `padding` (Polsterung). Das »padding«, der Innenabstand, gehört zum Innenbereich der Box, wo auch der Inhalt zu finden ist. Der »border« legt sich um das »padding« und den Inhalt und hat eine eigene Breite (`-width`), Farbe (`-color`) und Linienart (`-style`).



Sowohl für den »padding« als auch für den »border« kannst du vier verschiedene Werte angeben – für die vier Seiten der Box. Wie beim »margin« kannst du auch alle vier Werte in eine Zeile schreiben.

Die Eigenschaft, um die es geht, wird einfach angehängt.
Beispiele sind:

- » border-right-color
- » border-top-style
- » border-width



Für den »color«-Wert kannst du wieder eine vordefinierte Farbe nutzen und den »width«-Wert gibst du wieder in Pixeln an.

Für die »style«-Eigenschaft gibt es auch vordefinierte Werte.
Mit `none` bleibt die Linie unsichtbar, mit `dotted` erscheint sie gepunktet, mit `dashed` gestrichelt, mit `solid` durchgezogen und `double` sorgt für eine doppelte Linie.

`none` `dotted` `dashed` `solid` `double`

DAS RICHTIGE WERKZEUG

Sicher ist es möglich, einen Nagel mithilfe eines Nudelholzes in eine Wand zu schlagen. Allerdings erleichtert ein Hammer diese Aufgabe doch sehr.

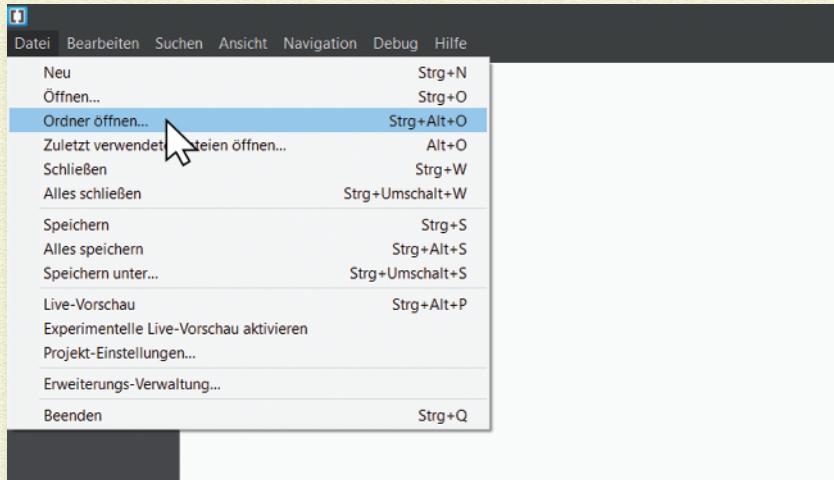
Dementsprechend solltest du auch bei der Webseitenentwicklung Wert darauf legen, das richtige Werkzeug zur Hand zu haben: Ein Code-Editor macht das Erstellen der Webseite sehr viel einfacher als ein normaler Texteditor. Ein guter Code-Editor für HTML und CSS ist *Brackets*.

Auf der Webseite brackets.io kannst du Brackets herunterladen und danach installieren, indem du mit Doppelklick die heruntergeladene Datei öffnest.

START MIT BRACKETS

1 Öffne Brackets.

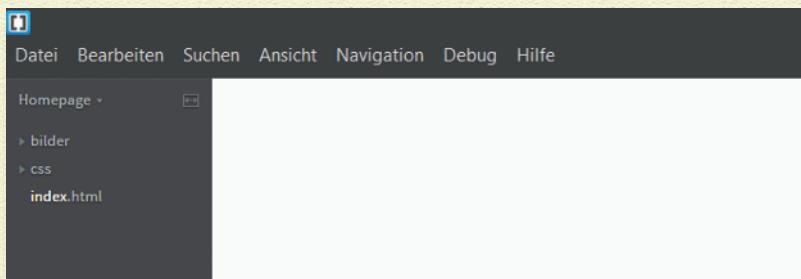
2 Klicke auf Datei und dann auf Ordner öffnen.



Es öffnet sich ein Dialogfenster.

3 Navigiere zum Ordner, in dem deine HTML-Datei liegt, und klicke auf Ordner auswählen.

Auf der linken Seite siehst du dein Webseitenprojekt und die Unterordner »bilder« und »css«.



4 Wenn du deine HTML-Datei anklickst, öffnet sich diese und du kannst sie bearbeiten.

Und was sind jetzt die Vorteile von Brackets? Da gibt es eine ganze Reihe:

- » Mit einem Klick auf **Datei** und dann auf **Live-Vorschau** öffnet sich die Webseite im Browser. *Live-Vorschau* bedeutet, dass du Änderungen, die du im Dokument gemacht hast, im Browser sofort anschauen kannst. Die Änderungen musst du dafür nicht einmal speichern. Und **F5** musst du auch nicht drücken.
- » Wenn du mit dem Mauszeiger über den Dateipfad eines Bildes oder den Namen einer Farbe rübergehst, wird **eine kleine Vorschau** davon im Editor angezeigt.



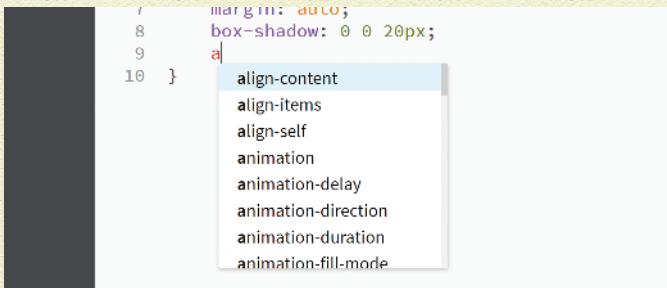
- » Außerdem siehst du auf deiner Webseite immer **einen dünnen Rahmen** um den Bereich, den du im Editor gerade bearbeitest. Wenn der Code mal länger wird, ist das eine große Hilfe.
- » Eine weitere eingebaute Funktion von Brackets ist die sogenannte **Syntaxhervorhebung**. Das heißt, Tags werden mit einer anderen Farbe eingefärbt als der Inhalt. HTML-Attribute haben wiederum eine eigene Farbe, die sich von der Tag-Farbe unterscheidet, und genauso ihre Werte. Somit lassen sich die verschiedenen Elemente leicht voneinander unterscheiden.

 A screenshot of the Brackets code editor showing a simple HTML file. The code is:


```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Titel der Webseite</title>
<link rel="icon" href="bilder/icon.ico">
<link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
<articles>
```

 The code uses color-coded syntax highlighting: tags like <html> and <head> are dark blue; attributes like "width=device-width" are orange; and values like "1.0" are light blue. The Brackets interface is visible around the code editor.

- » Sobald du innerhalb eines Selektors nur einen Buchstaben getippt hast, werden dir schon CSS-Eigenschaften vorgeschlagen, aus denen du dann auswählen kannst. Genauso werden Werte für die CSS-Eigenschaften angezeigt, sobald du »:« getippt hast.



Darüber hinaus kannst du sogar noch Erweiterungen von Brackets installieren, sodass du noch entspannter programmieren kannst.

1 Gehe auf Datei und dann auf Erweiterungs-Verwaltung.

Es öffnet sich ein Fenster, in dem ein paar Erweiterungen aufgelistet sind. Oben rechts existiert auch eine Suchleiste, sodass du nach Erweiterungen suchen kannst.

2 Gib dort Color Picker oder Beautify ein.

Neben den Suchergebnissen erscheint rechts ein kleiner Button **Installieren**. Wenn du den drückst, installiert sich die Erweiterung und du kannst sie nutzen.

Mit **Color Picker** (Farbwähler) kannst du aus einem Farbfeld eine beliebige Farbe wählen und hast somit eine Erweiterung zu den vordefinierten Farben. Die ausgewählte Farbe wird dann in deinem Programm im Hexadezimalformat eingetragen.



Mit **Beautify** (Verschönern) kannst du deinen gesamten Code markieren und mit einem Klick auf **Beautify** unter **Bearbeiten** wird dein Code ordentlich formatiert, also Zeilen werden eingerückt oder Leerzeilen eingefügt.

Es gibt noch weitere unzählige Erweiterungen für Brackets und es lohnt sich, danach zu googeln und dann zu entscheiden, was dir das Arbeiten erleichtern würde.

BROWSER-SUPPORT IST ANSICHTSSACHE

Jeder Browser hat ein eigenes eingebautes Stylesheet, welches sich von dem Stylesheet eines anderen Browsers unterscheidet. Leider können nicht alle Browser jede CSS-Eigenschaft umsetzen.

Manche Browser benötigen für bestimmte CSS-Eigenschaften eine Vorsilbe, der Profi sagt ein Browserpräfix, damit die Eigenschaft angezeigt werden kann. Solche Präfixe können so aussehen:

- » -moz-
- » -webkit-
- » -ms-

Diese Präfixe schreibt man – wenn nötig – vor die CSS-Eigenschaft.

Die Unterstützung von CSS-Eigenschaften ändert sich von Browerversion zu Browerversion. Der Code im Buch wurde mit Google Chrome Version 58.0.3029.81 und Mozilla Firefox Version

53.0 getestet. Mit diesen Browserversionen wurden auch die Screenshots erstellt.



Der Internet Explorer unterstützt nicht alle Befehle, die wir in diesem Buch benutzen. Bitte benutze deshalb lieber einen anderen Browser wie Chrome oder Firefox.

Am besten arbeitet man mit CSS-Eigenschaften, die von allen Browsern unterstützt werden, um Darstellungsunterschiede in verschiedenen Browsern zu vermeiden. In diesem Buch lernt ihr nur solche CSS-Eigenschaften kennen.

CODE

Hier findest du den Code des `index.html`-Dokuments, welches wir in diesem ersten Kapitel erstellt haben:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Titel der Webseite</title>
    <link rel="icon" href="bilder/icon.ico">
    <link rel="stylesheet" type="text/css"
        href="css/style.css">
</head>

<body>
    <article>
        <header>
        </header>
        <section>
            Überraschung
        </section>
    </article>
</body>

</html>
```

Hier findest du den Code der dazugehörigen `style.css`-Datei:

```
body{  
    margin: 0;  
}  
  
article{  
    margin: auto;  
    width: 1100px;  
    box-shadow: 0 0 20px;  
}
```