

CSS



HTML



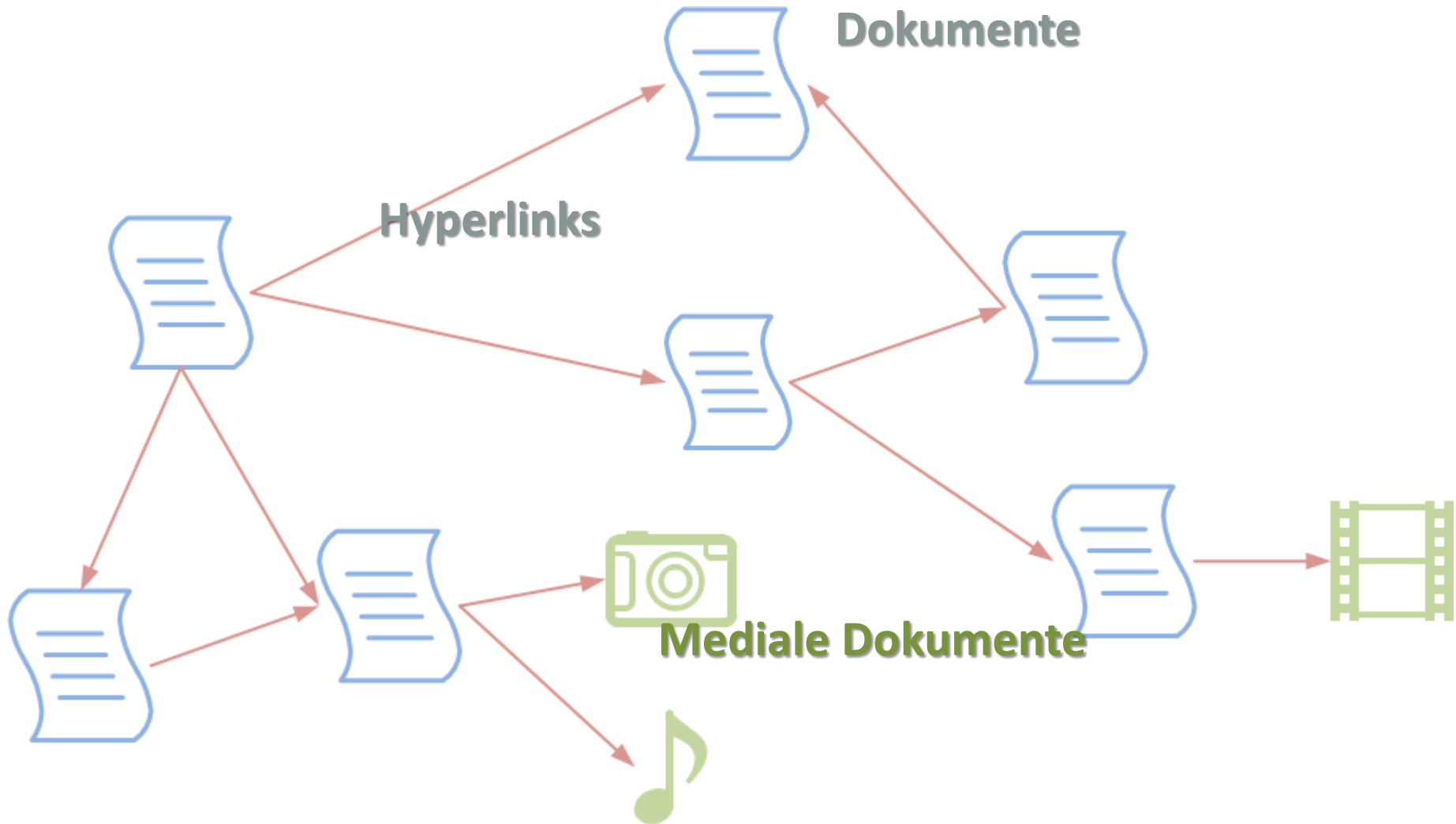
WEB-TECHNOLOGIEN

Grundlegende HTML-Elemente – Teil 2

Grundlegende HTML-Elemente

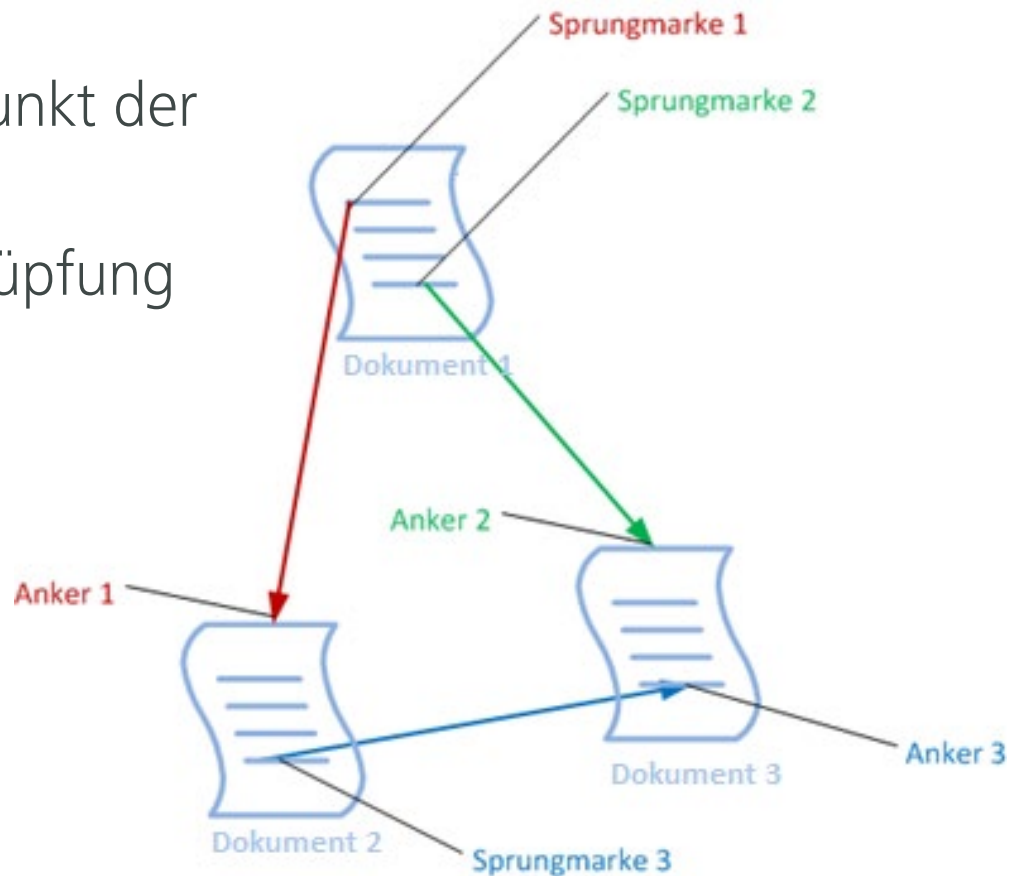
- › Elemente zur Strukturierung von Dokumentenabschnitten
- › Elemente zur Hervorhebung von Texten
- › Links
- › Bilder

Hypertext-Dokumente und Hyperlinks



HTML-Links: Wichtigste Elemente und Attribute

- › Hypertexte werden über zwei Informationen miteinander verknüpft:
 - › Sprungmarke: Startpunkt der Verknüpfung
 - › Anker: Ziel der Verknüpfung



Beispiel: Links aus dem „Dokument 1“

```
<h1>Dokument 1</h1>
```

```
<p>
```

Das ist die erste

```
<a href="Dokument2.html">Sprungmarke</a>.
```

Sie verlinkt dieses Dokument mit dem "Dokument2".

```
</p>
```

```
<p>
```

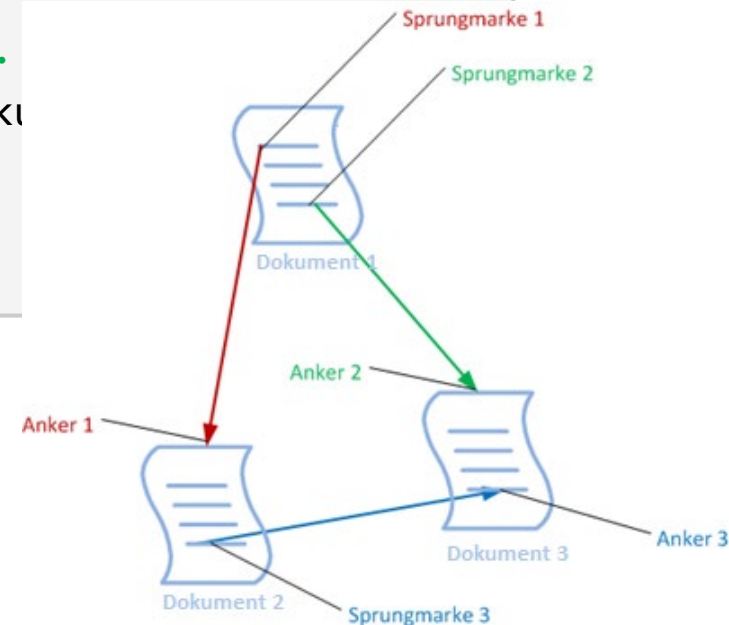
Das ist die zweite

```
<a href="Dokument3.html">Sprungmarke</a>.
```

Sie verlinkt dieses Dokument mit dem "Doku

```
</p>
```

Beispiel



HTML-Links: Wichtigste Elemente und Attribute

- › Die Links werden in HTML an den Stellen markiert, an denen ein Dokument mit einem anderen Dokument verknüpft werden soll (*Sprungmarke*).
- › Die Verknüpfung erfolgt durch die Angabe des *Ankers* im Zielement.
- › Element: `<a>` (*anchor*)
 - › Attribute des Elementes `<a>`:
 - › `href`: Angabe der Zieladresse (URL)
 - › Kinder des Elementes `<a>`:
 - › Text, der im Browser an der Sprungmarke angezeigt wird.

HTML-Links: Angabe der Zieladresse

- › **href**: Angabe der *Zieladresse*
- › Die Zieladresse wird über einen *relativen Pfad* angegeben.
- › *Was sind aber relative Pfade?*

Exkurs: Betriebssysteme - Dateitypen

- › Die meisten Betriebssysteme unterscheiden zwischen zwei Typen von Dateien:
 - › Reguläre Dateien
 - › Verzeichnisse

Exkurs: Betriebssysteme – Reguläre Dateien

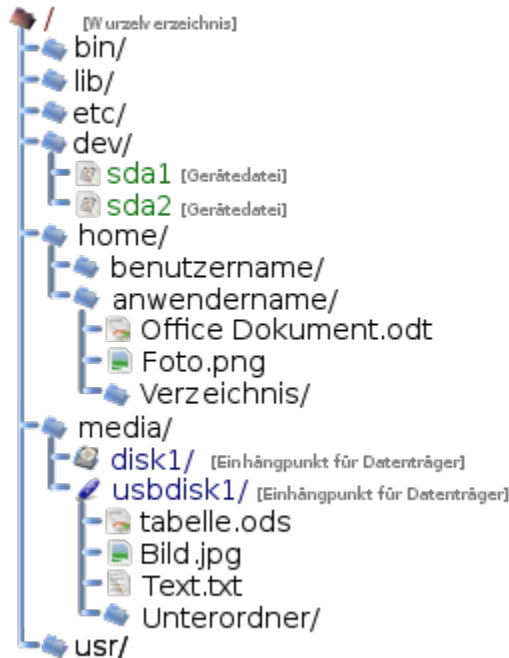
- › bestehen aus mehreren Zeilen unterschiedlicher Länge
- › können mit Hilfe von Programmen angezeigt und bearbeitet werden
- › können beliebige Namen haben:
 - › meistens: **<Name>.<Endung>**
 - › Die Endung (Typangabe) dient meistens der Zuordnung zum Bearbeitungs- oder Ansichtsprogramm

Exkurs: Betriebssysteme - Verzeichnisse

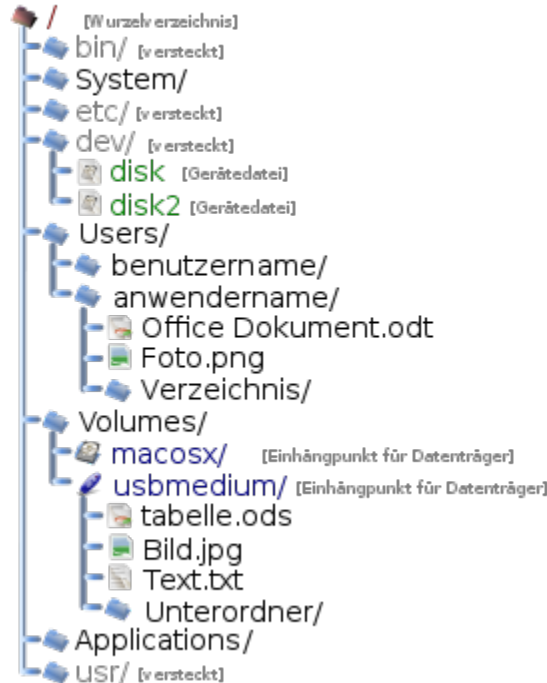
- › Verzeichnisse dienen der Verwaltung mehrerer Dateien
- › Innerhalb einer Verzeichnisstruktur existiert für jede Datei ein eindeutiger Zugriffspfad
- › Dateien innerhalb eines Verzeichnisses müssen unterschiedliche Namen haben
- › Verzeichnisstrukturen sind meistens hierarchisch in einer Baumstruktur angelegt

Exkurs: Betriebssysteme - Verzeichnisse

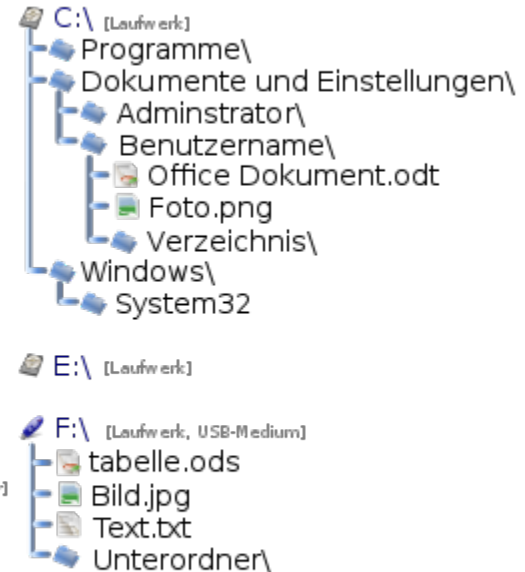
UNIX, Linux, ZETA



Mac OSX



Windows NT/2000/XP

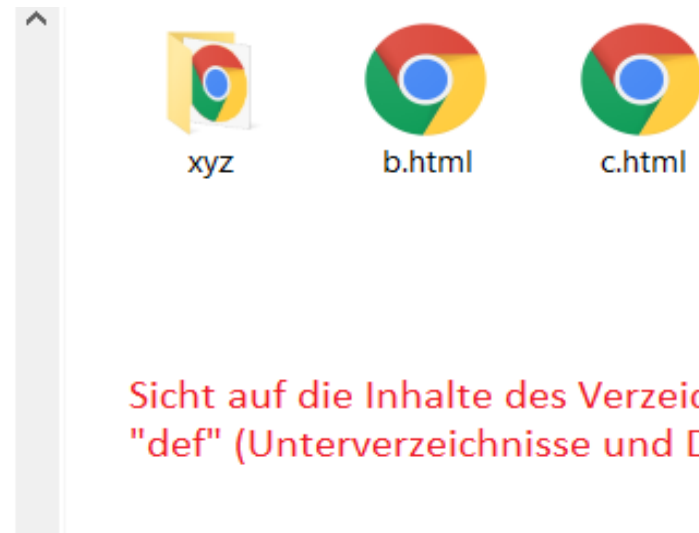


[aus: <http://de.wikipedia.org/w/index.php?title=Datei:Filesystem.svg&filetimestamp=20100727221414&>]

Sicht auf Verzeichnisse und Dateien im Windows Explorer

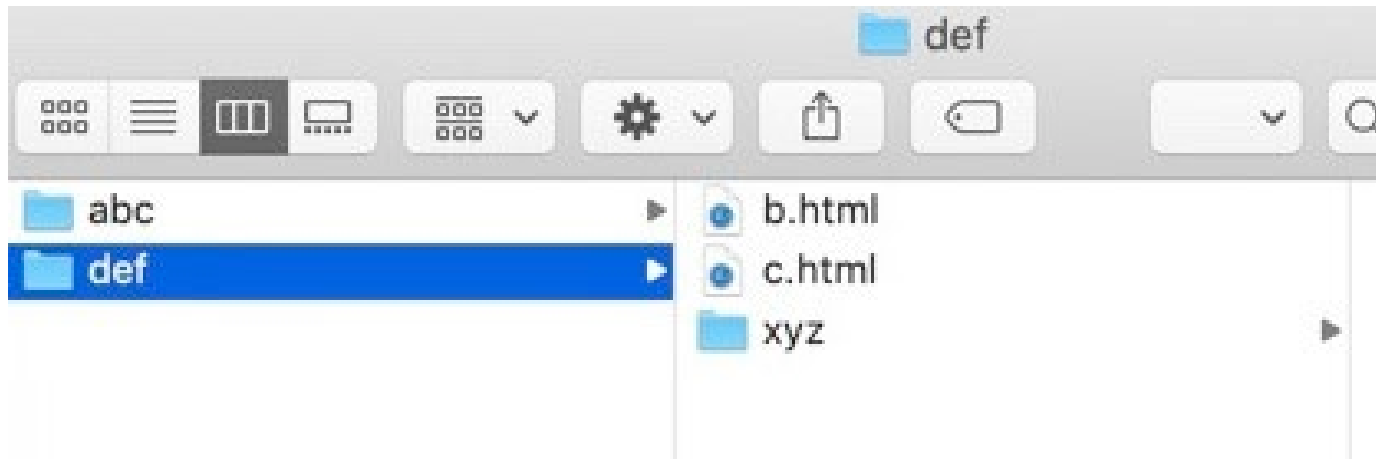


Sicht auf die Verzeichnisstruktur



**Sicht auf die Inhalte des Verzeichnisses
"def" (Unterverzeichnisse und Dateien)**

Sicht auf Verzeichnisse und Dateien im Finder



Sicht auf die Verzeichnisse

Sicht auf die Inhalte des Verzeichnisses
„def“ (Unterverzeichnisse und Dateien)

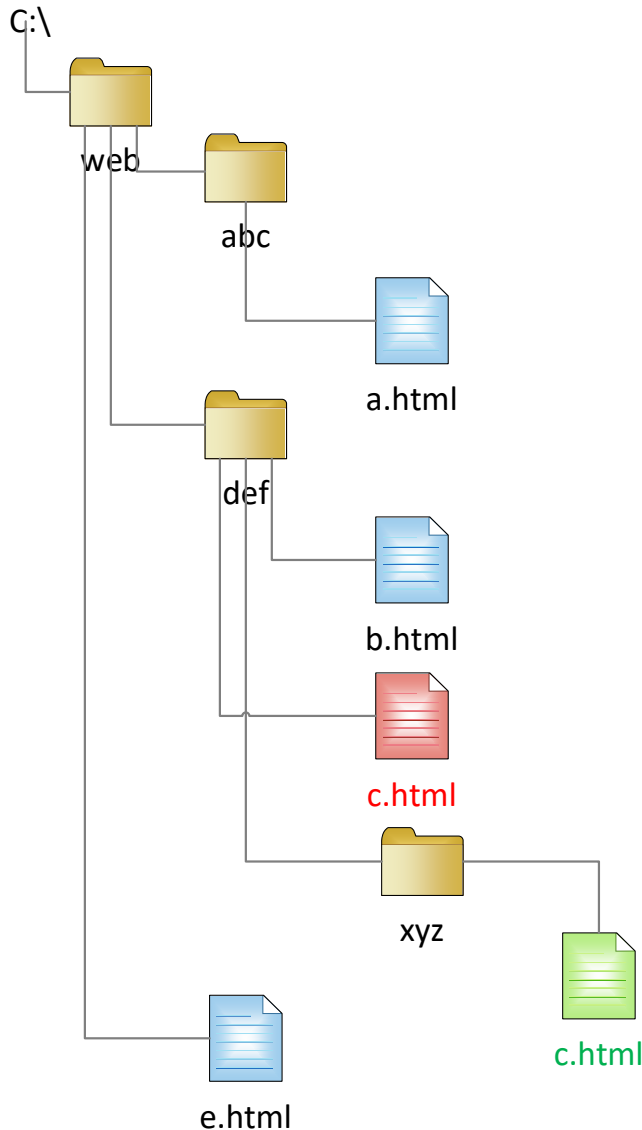
Exkurs: Betriebssysteme - Absolute Pfade

- › Zum Zugriff auf eine Datei innerhalb einer hierarchischen Verzeichnisstruktur werden *Pfade* benutzt.
- › Ein *absoluter Pfadname* spezifiziert den gesamten Pfad vom Wurzelverzeichnis bis zur Datei.

Exkurs: Betriebssysteme - Relative Pfade

- › Ein *relativer Pfad* spezifiziert den Pfad zu einer Datei ausgehend vom „aktuellen Verzeichnis“. Das „aktuelle Verzeichnis“ ist das Verzeichnis, auf dem das ausgeführte Programm gerade arbeitet.
- › *Im Web werden ausschließlich relative Pfade benutzt.*

Unterschied: Absolute – relative Pfade



Absoluter Pfad zur Datei **c.html** im Verzeichnis **def**:
C:/web/def/c.html

Absoluter Pfad zur Datei **c.html** im Verzeichnis **xyz**:
C:/web/def/xyz/c.html

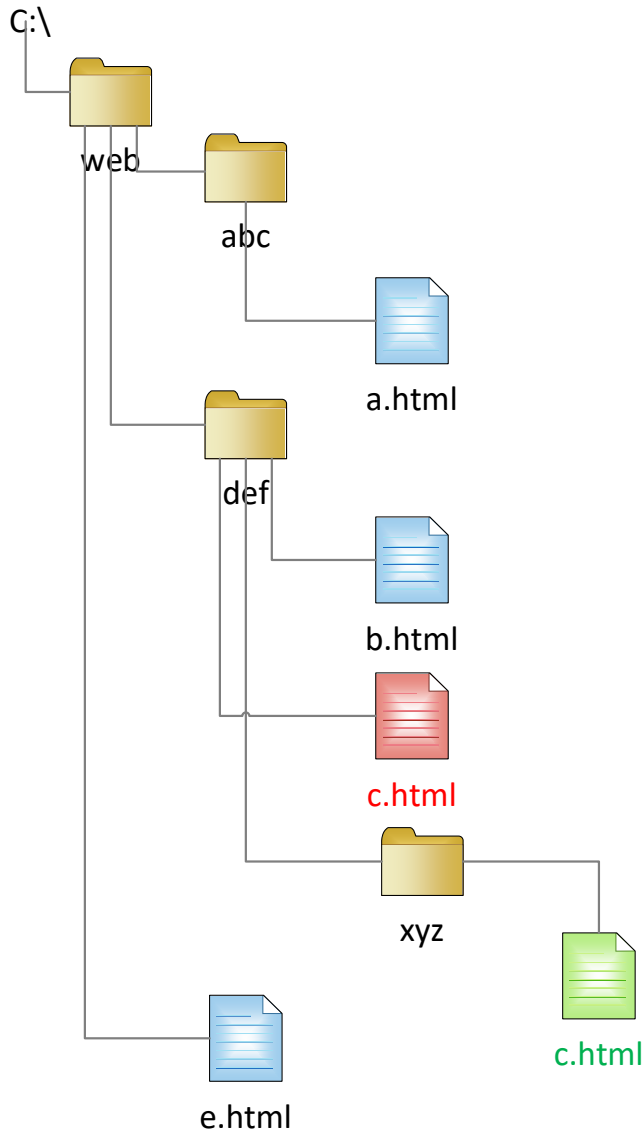
Relativer Pfad von der Datei **e.html** zur Datei
c.html im Verzeichnis **def**:
def/c.html

Relativer Pfad von der Datei **a.html** zur Datei
c.html im Verzeichnis **xyz**:
../def/xyz/c.html

Syntax der Angabe relativer Pfade

- › Mit **./** wird das Verzeichnis gekennzeichnet, in dem sich das Ausgangsdokument befindet. Folgende Angaben relativer Pfade sind daher synonym:
 - › **c.html**
 - › **./c.html**
- › Mit **/** wird ein Unterverzeichnis referenziert:
 - › **def/c.html**
- › Mit **../** wird das übergeordnete Verzeichnis referenziert:
 - › **../e.html**
- › Diese Angaben können beliebig miteinander verschachtelt werden:
 - › **../../abc/a.html**

Syntax der Angabe relativer Pfade



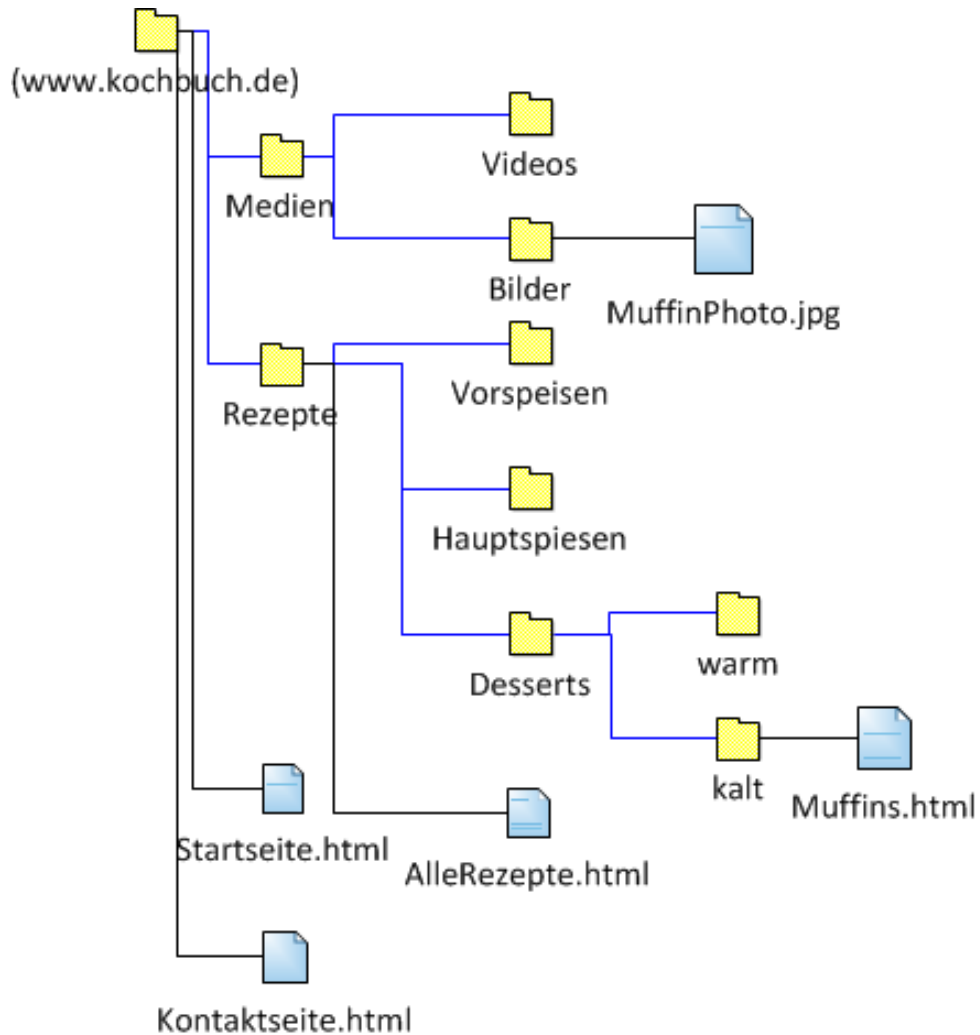
Relativer Pfad von der Datei **e.html** zur Datei **c.html** im Verzeichnis **def**:
def/c.html

Relativer Pfad von der Datei **e.html** zur Datei **c.html** im Verzeichnis **xyz**:
def/xyz/c.html

Relativer Pfad von der Datei **b.html** zur Datei **c.html** im Verzeichnis **def**:
c.html oder **./c.html**

Relativer Pfad von der Datei **a.html** zur Datei **c.html** im Verzeichnis **xyz**:
../def/xyz/c.html

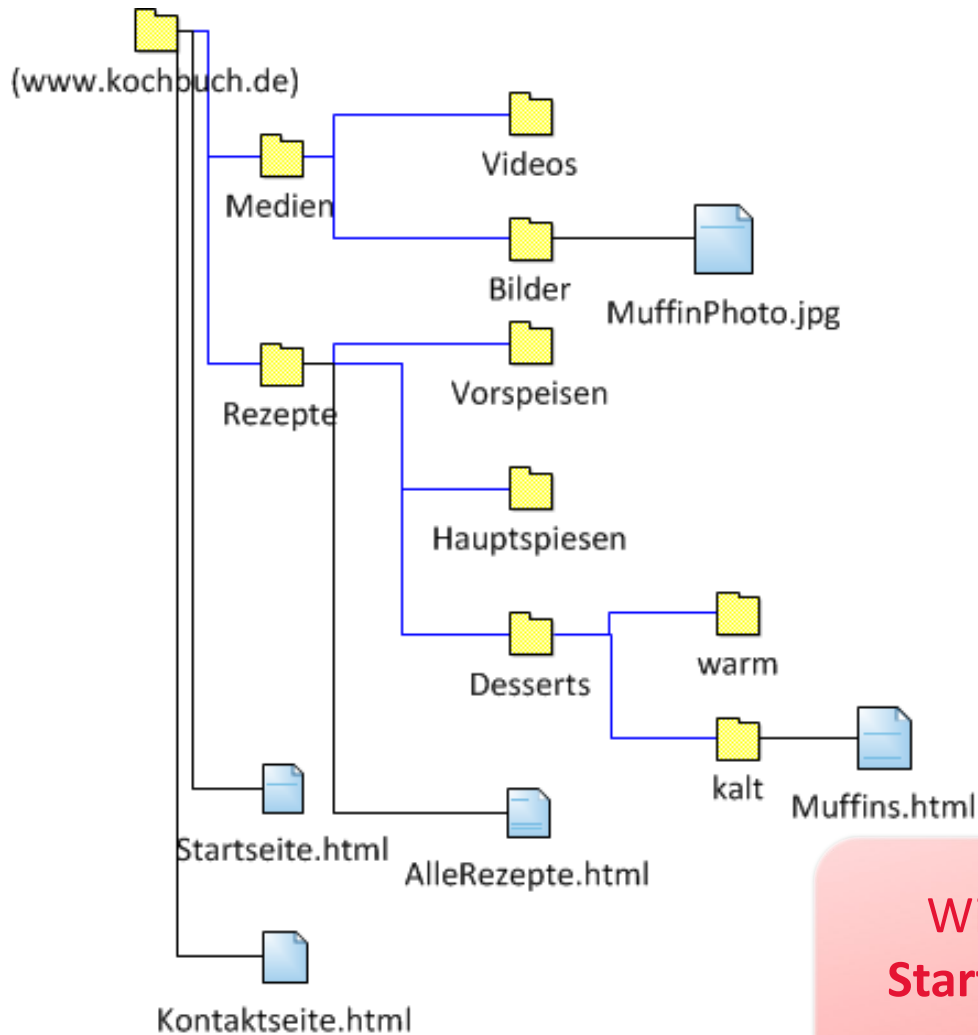
Beispiel einer Verzeichnisstruktur



HTML: Relative Pfade

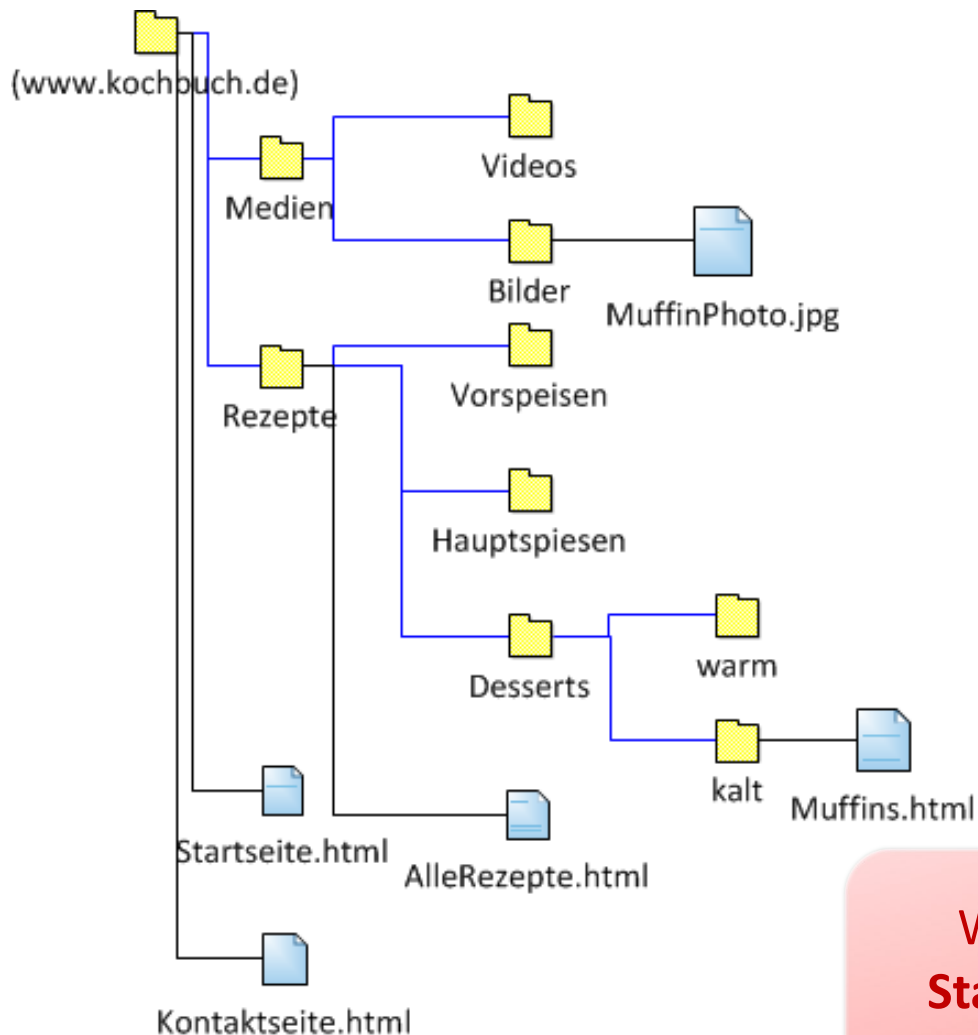
<https://e-learning.hdm-stuttgart.de/moodle/mod/quiz/view.php?id=279679>

Interne Zielangabe: Ziel im selben Verzeichnis



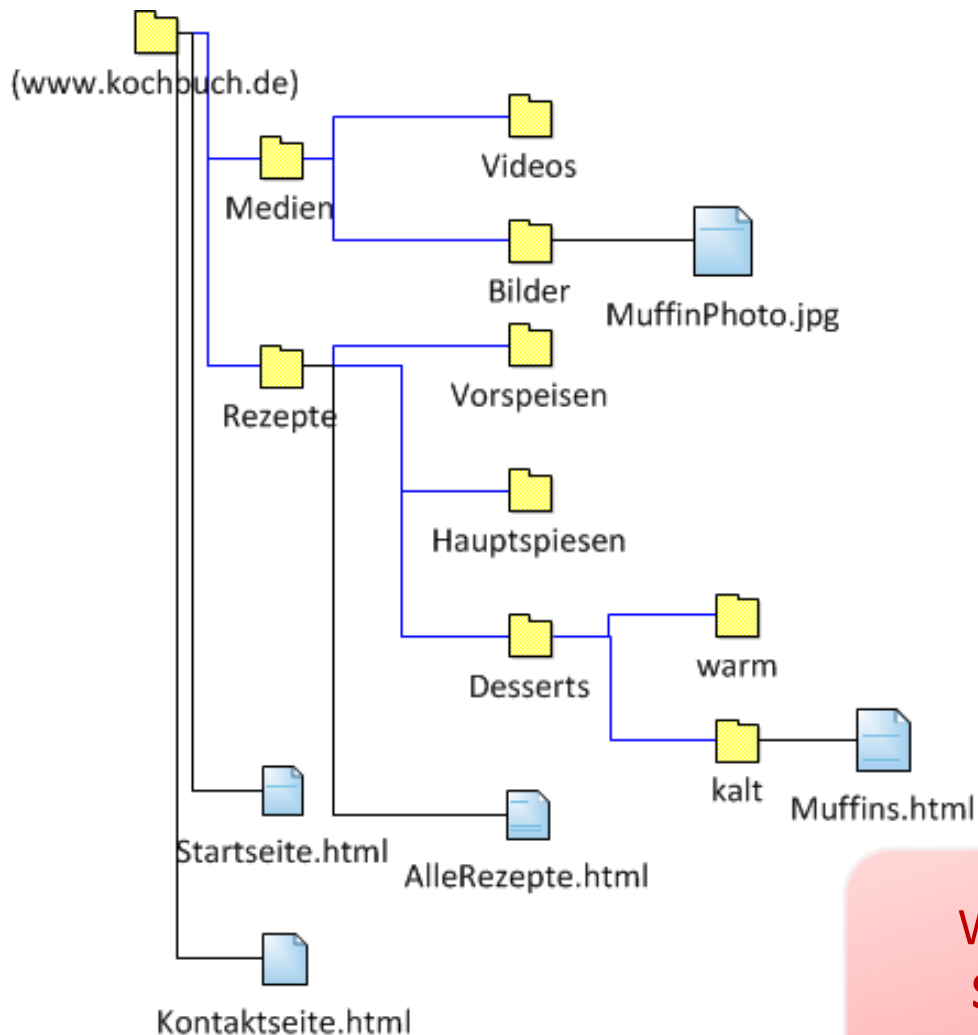
Wie wird der relative Pfad von der **Startseite.html** zur **Kontaktseite.html** definiert?

Interne Zielangabe: Ziel im untergeordnetem Verzeichnis



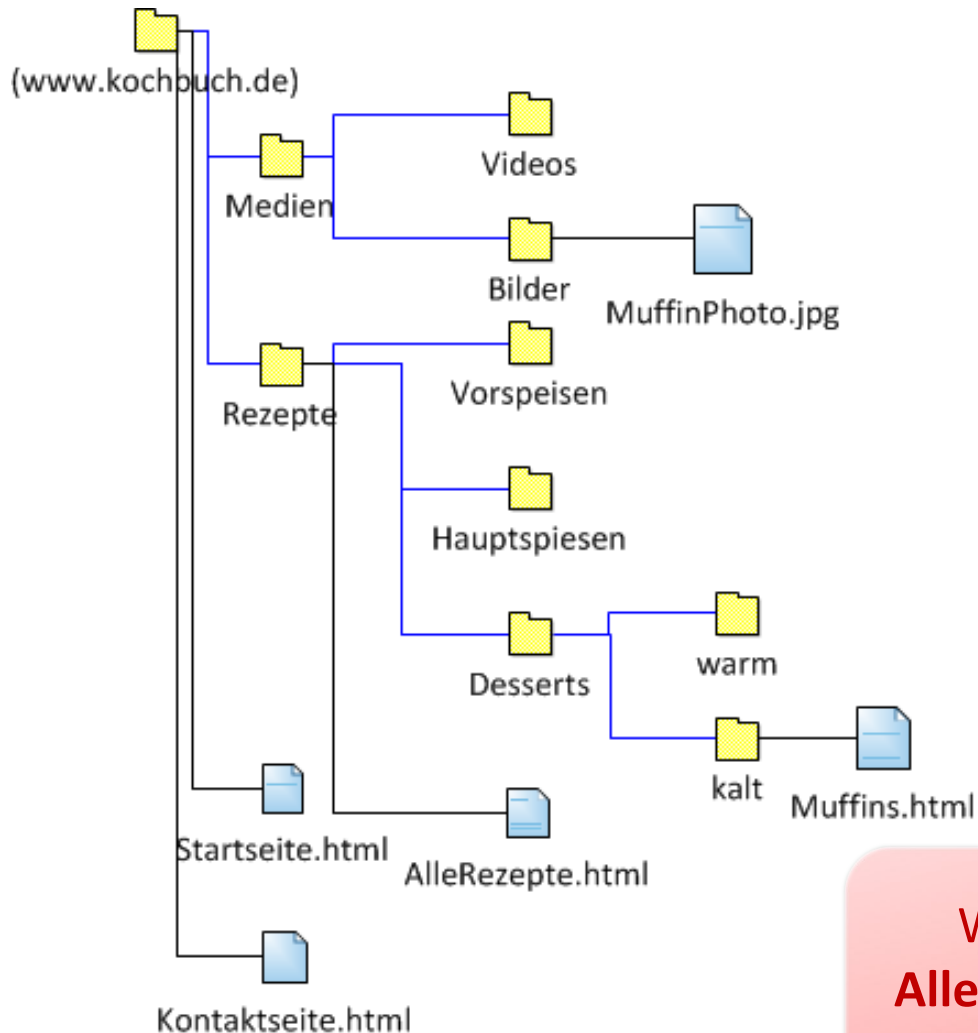
Wie wird der relative Pfad von der **Startseite.html** zur **AlleRezepte.html** definiert?

Interne Zielangabe: Ziel in untergeordneter Verzeichnisstruktur



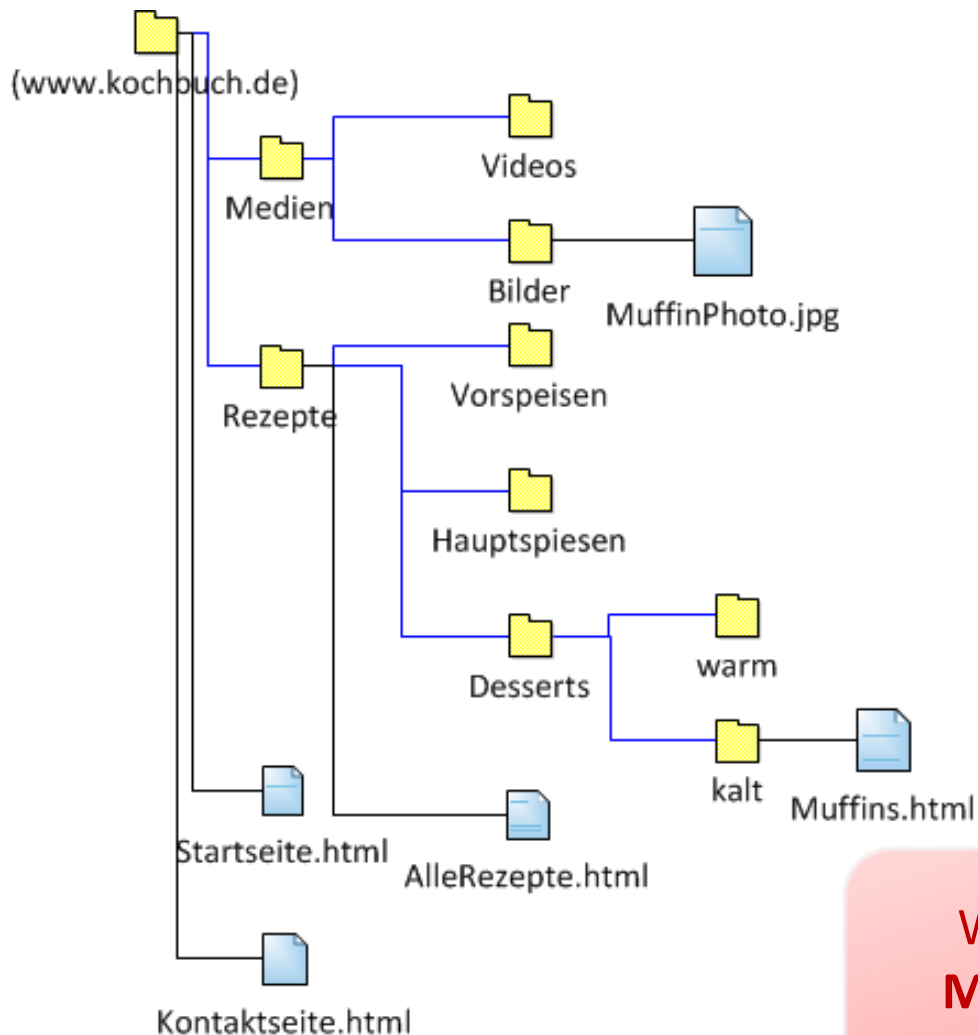
Wie wird der relative Pfad von der **Startseite.html** zur **Muffins.html** definiert?

Interne Zielangabe: Ziel im übergeordneten Verzeichnis



Wie wird der relative Pfad von der **AlleRezepte.html** zur **Kontaktseite.html** definiert?

Interne Zielangabe: Ziel in übergeordneter Verzeichnisstruktur



Wie wird der relative Pfad von der **Muffins.html** zum **MuffinPhoto.jpg** definiert?

HTML-Links: Angabe der Zieladresse

- › Die Zieladresse wird über einen *relativen Pfad* angegeben.
- › Externer Server: Ausgangsdatei und Zielfile befinden sich auf unterschiedlichen Rechnern
 - › Angabe der Serveradresse
 - › Angabe des relativen Pfads

```
http://www.hdm-stuttgart.de/wing/index.html
```

- › Interne Zieladresse: Ausgangsdatei und Zielfile befinden sich auf demselben Rechner
 - › Angabe des relativen Pfads

```
Rezepte/AlleRezepte.html
```

HTML-Links: Beispiel einer internen Zielangabe

```
<h1>Dokument 1</h1>
```

```
<p>
```

Das ist die erste

```
<a href="Dokument2.html">Sprungmarke</a>.
```

Sie verlinkt dieses Dokument mit dem "Dokument2".

```
</p>
```

```
<p>
```

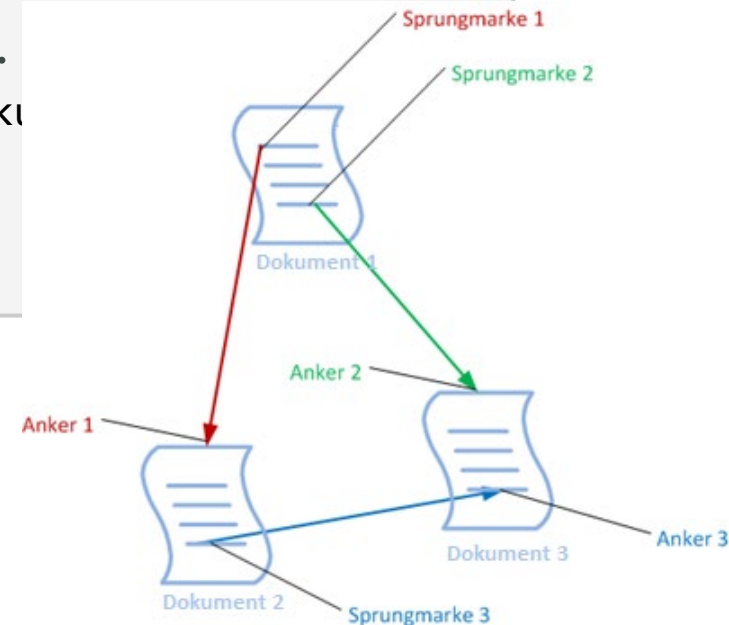
Das ist die zweite

```
<a href="Dokument3.html">Sprungmarke</a>.
```

Sie verlinkt dieses Dokument mit dem "Doku

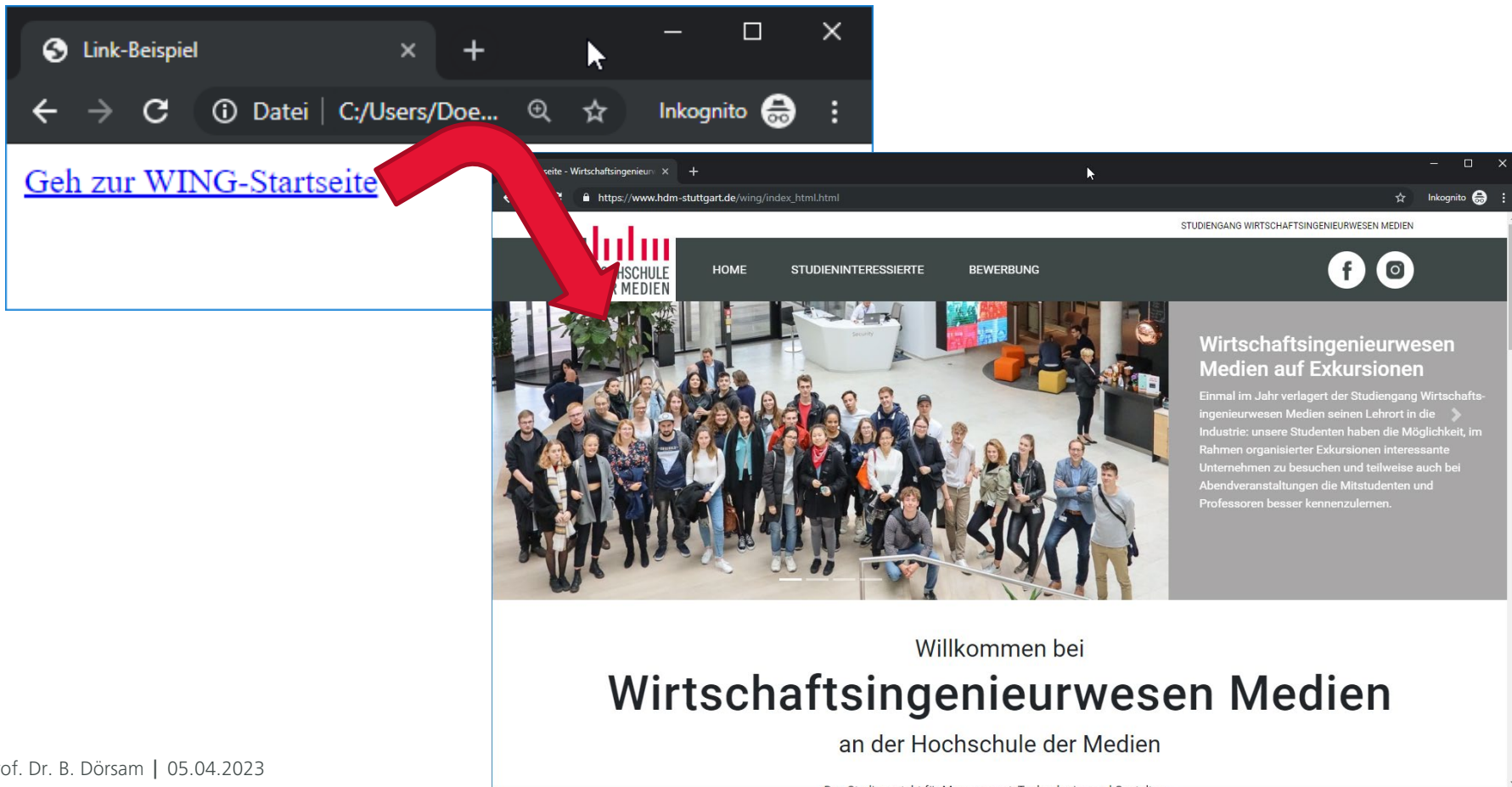
```
</p>
```

Beispiel



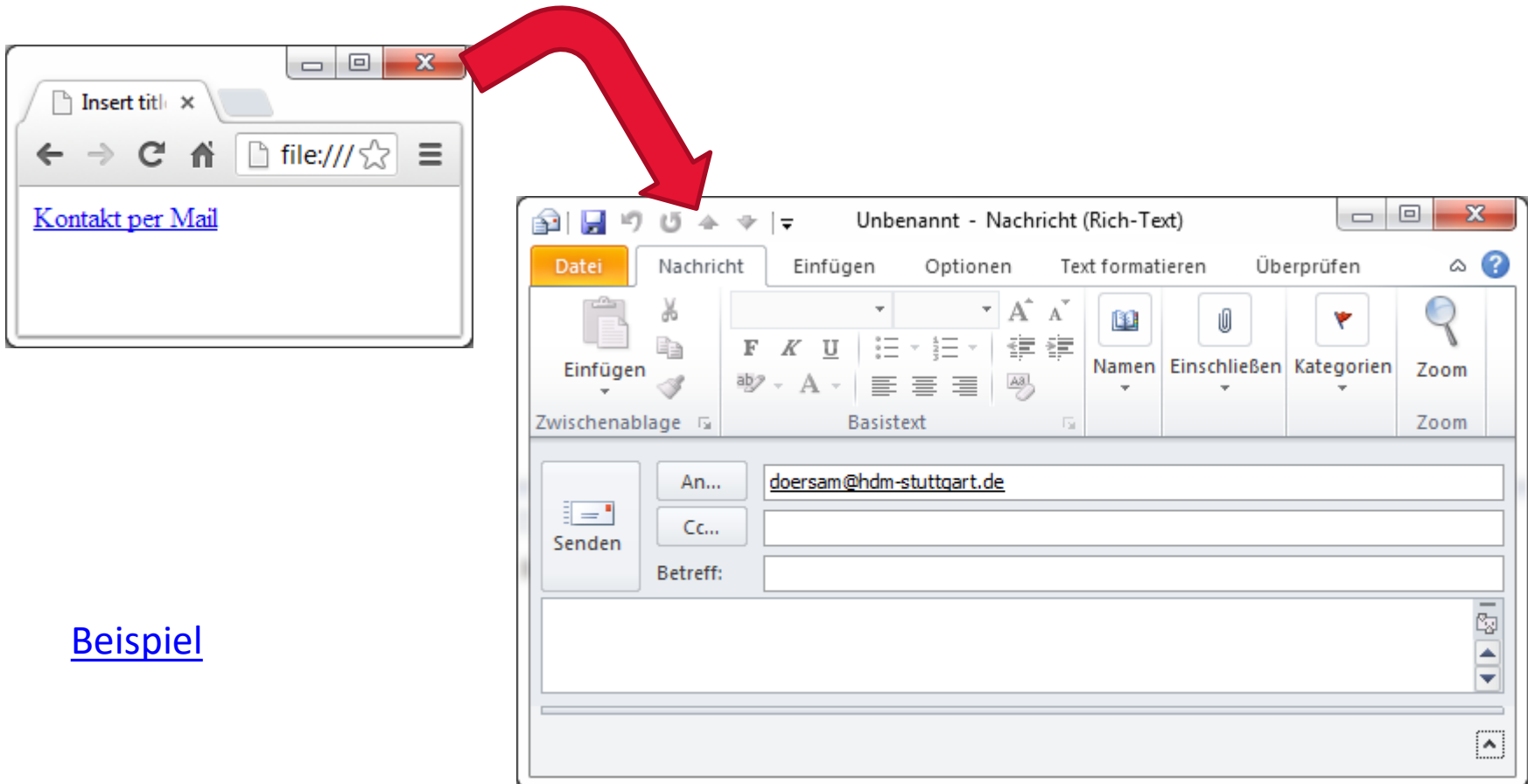
HTML-Links: Beispiel einer externer Zielangabe

```
<a href="http://www.hdm-stuttgart.de/wing/index_html.html">  
    Geh zur WING-Startseite  
</a>
```



Sonderfall: Mail-Link

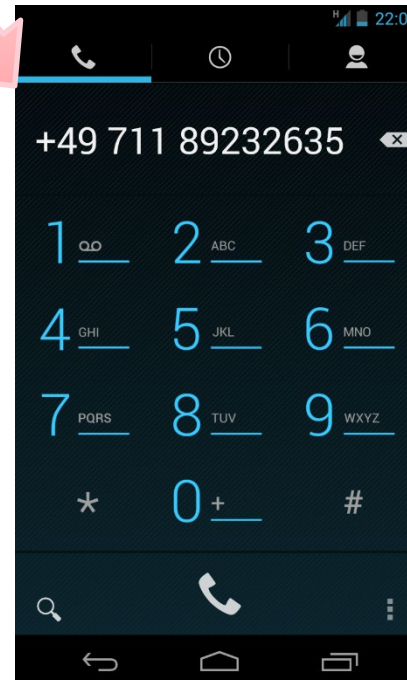
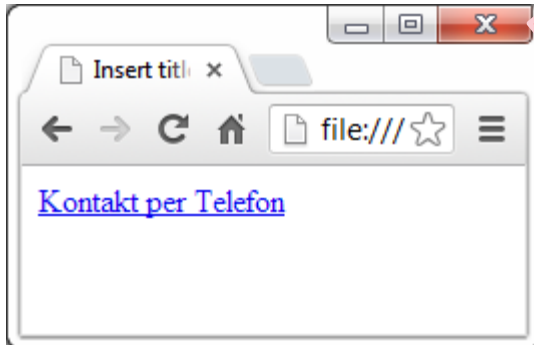
```
<a href = "mailto:doersam@hdm-stuttgart.de">Kontakt per Mail</a>
```



Beispiel

Sonderfall: Telefonlink

```
<a href="tel:+4971189232635">Kontakt per Telefon</a>
```

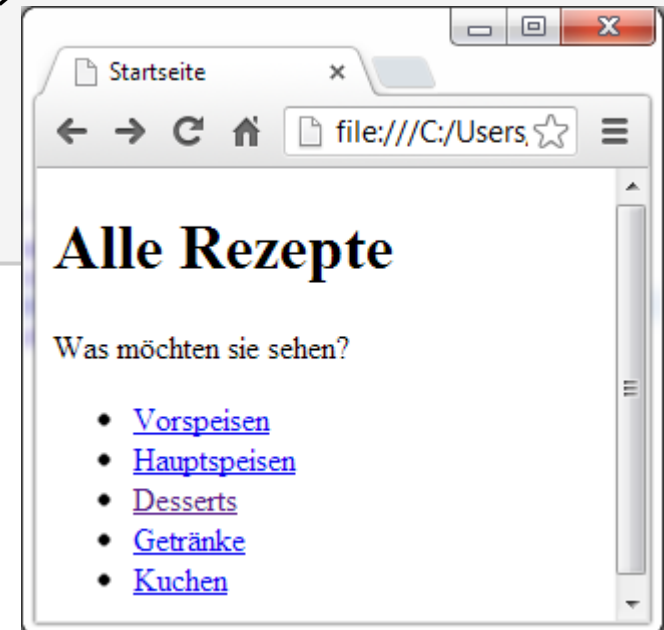


Beispiel

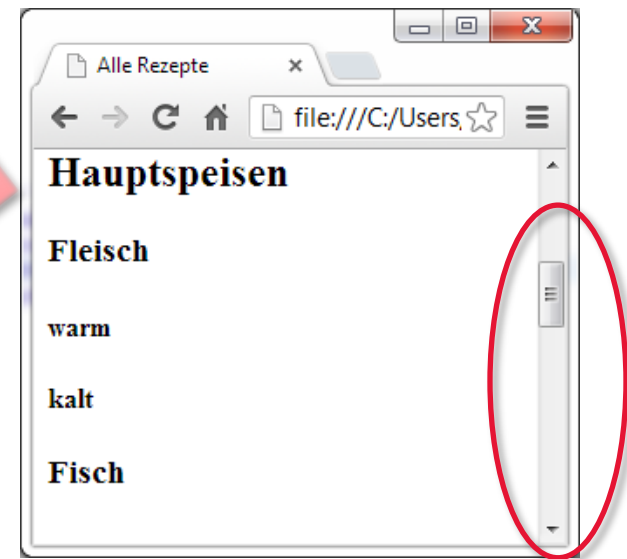
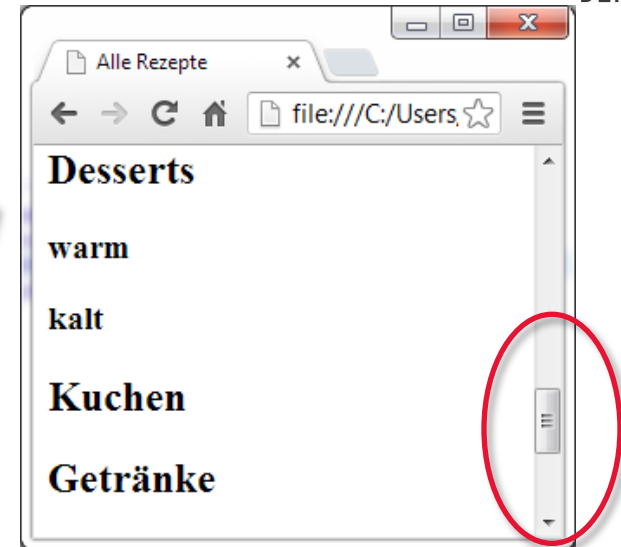
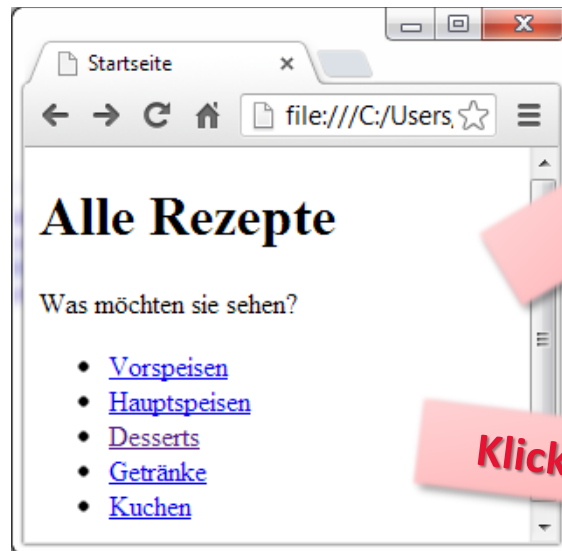
Links auf spezielle Stellen im Dokument

```
<ul>  
  ...  
  <li>  
    <a href="AlleRezepte.html#hauptspeisen">Hauptspeisen</a>  
  </li>  
  <li>  
    <a href="#desserts">Desserts</a>  
  </li>  
  ...  
</ul>
```

Beispiel



Links auf spezielle Stellen im Dokument



Links auf spezielle Stellen im Dokument

- › Um Links auf spezielle Stellen innerhalb eines Dokumentes zu setzen, müssen diese *speziell gekennzeichnet* werden.
- › Die Notwendigkeit, eine Stelle im Dokument eindeutig zu kennzeichnen gibt es auch für viele andere Anwendungsfälle.
- › Deswegen gibt es in HTML die Möglichkeit, *fast jedes HTML-Element* durch ein so genanntes *Universalattribut id* eindeutig zu kennzeichnen.

Das Universalattribut **id**

- › **id** ist ein *datei-weit eindeutiger* Name für ein HTML-Element.
- › Es wird allerdings nicht automatisch überprüft, ob der Wert des Attributs **id** innerhalb einer Datei tatsächlich eindeutig ist.
- › Wird eine HTML-Datei manuell erstellt, so hat der Ersteller die Verantwortung dafür, keinen Wert innerhalb einer HTML-Datei mehr als einmal als **id** zu vergeben!

```
<h2 id="hauptspeisen">Hauptspeisen</h2>  
<h2 id="desserts">Desserts</h2>
```

Links auf einen **id**-Wert

- › Soll ein Link auf ein mit einer **id** gekennzeichnetes Element verweisen, so muss die Linkadresse um eine spezielle Angabe der Id erweitert werden.
- › Die Id wird innerhalb von Link-Elementen mit einem **#** eingeleitet.

```
<a href="AlleRezepte.html#hauptspeisen">Hauptspeisen</a>  
<a href="#desserts">Desserts</a>  
<a href="#">Start</a>  
<a href="AlleRezepte.html#">Start</a>
```

Grundlegende HTML-Elemente

- › Elemente zur Strukturierung von Dokumentenabschnitten
- › Elemente zur Hervorhebung von Texten
- › Links
- › Bilder

Texte strukturieren: Bilder

- › Bilder werden – anders als in Textverarbeitungsprogrammen wie MS Word – nicht in ein HTML-Dokument integriert, sondern lediglich *referenziert*.
- › Dafür steht das Element `<img>` (*image*) zur Verfügung.
- › Es handelt sich um ein leeres Element, d.h., das Element `<img>` hat keine Kinder, auch keinen Text.

Texte strukturieren: Bilder

- › Die wichtigsten Informationen über ein Bild stehen in den Attributen des Elements `<img>`:
 - › **src**: erforderliche Angabe der URL (Syntax analog zu Links)
 - › Extern, z.B.: <http://www.doersam.de/muffins.jpg>
 - › Lokal, z.B.: [muffins.jpg](#) oder [bilder/muffins.jpg](#), ...
 - › **alt**:
 - › alternativer Text, der angezeigt wird, wenn das referenzierte Bild auf dem Server nicht gefunden wird
 - › Nicht alle Browser berücksichtigen allerdings den alternativen Text:
 - › Firefox und Internet Explorer zeigen den alternativen Text an.
 - › Chrome und Safari zeigen den alternativen Text nicht an.
 - › **title**: Text, der angezeigt wird, wenn der Benutzer mit der Maus über das Bild fährt.

Texte strukturieren: Bilder

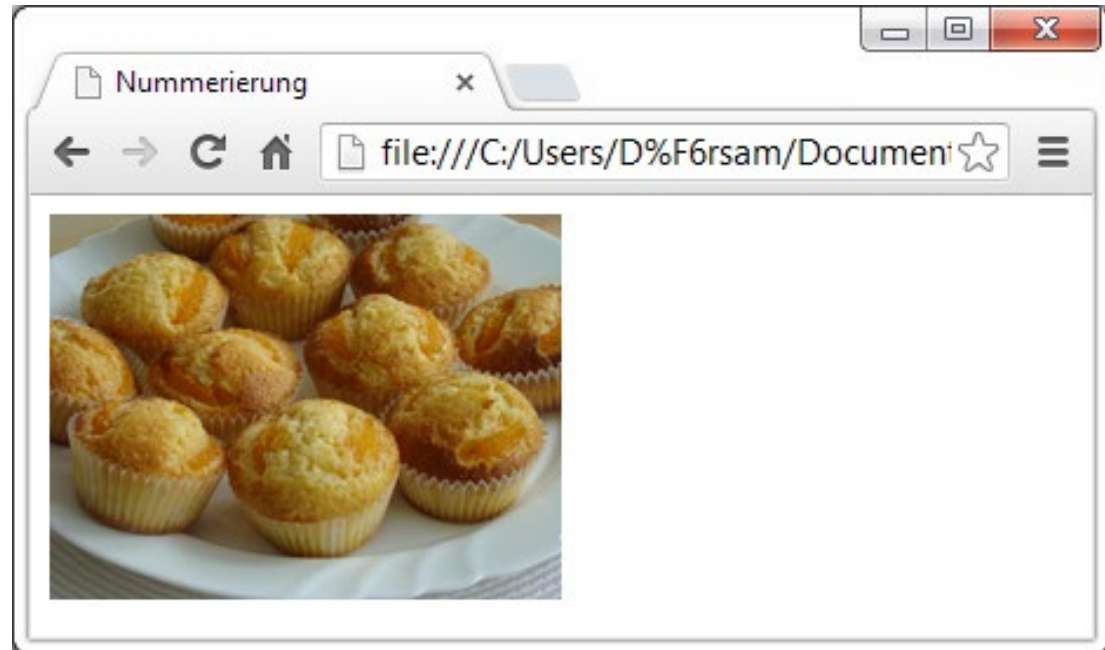
- › Weitere optionale Attribute von `<img>` sind `width` und `height`.
- › Beide können mit einer konkreten `Pixelzahl` benutzt werden:
 - › `<img src= "muffins.jpg" width="200" height="50">`
- › Werden beide Werte gesetzt, so wird das referenzierte Bild entsprechend skaliert.
- › Wird nur ein Wert gesetzt, so wird der zweite Wert vom Browser so berechnet, dass beide Kantenlängen des Bildes proportional skaliert werden.

Texte strukturieren: Bilder

```

```

Beispiel



Texte strukturieren: Bilder

Tipps zur Skalierung von Bildern:

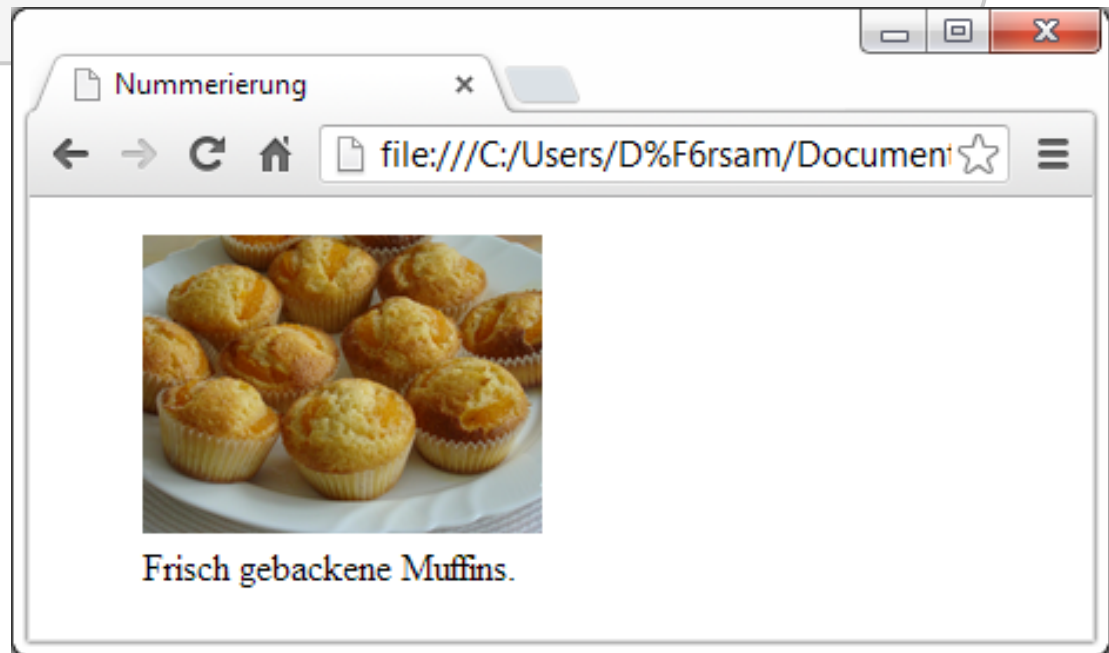
- Die Benutzung der Attribute `width` und `height` mit festen Pixelwerten beschleunigt den Aufbau einer Seite im Browser, da dadurch der Platzbedarf fürs Bild von vorne bekannt ist.
- Allerdings sind feste Bildgrößen ungeeignet, wenn eine Seite auf unterschiedlichen Ausgabegeräten (PC, Tablet, Smartphone) dargestellt werden soll.
- Die Breiten- und Höhenangaben sollten niemals größer sein als die Originalgröße des Bildes (keine Vergrößerung).
- Um die Ladezeiten der Bilder zu reduzieren, ist es oft besser, die Bilder selbst in einem Bildbearbeitungsprogramm zu verkleinern und dann auf einer HTML-Seite in Originalgröße einzubinden.

Texte strukturieren: Abbildungen

- › Bilder können auch – wie aus technischen und wissenschaftlichen Dokumenten bekannt – als Abbildungen behandelt werden.
- › Eine Abbildung (*figure*) zeichnet sich dadurch aus, dass sie
 - › eine Unter- oder Überschrift (*figure caption*) hat und
 - › teilweise auch anders formatiert wird als der restliche Text.
- › Deswegen gibt es in HTML5 zusätzlich die Elemente `<figure>` und `<figcaption>`.

Texte strukturieren: Abbildungen

```
<figure>
  <img src= "muffins.jpg" alt="Muffin-Bild" width="200" />
  <figcaption>
    Frisch gebackene Muffins.
  </figcaption>
</figure>
```



Unterschied: Bild - Abbildung

Texte strukturieren: Abbildungen

Bedeutung der Abbildungs-Elemente:

- Die Benutzung der Abbildungs-Elemente in HTML ist vor allem wichtig, wenn das Seitenlayout mit Hilfe von Stylesheets definiert wird.
- Eventuell können diese Elemente auch zur automatischen Bearbeitung von HTML-Seiten mit JavaScript benutzt werden.
- Auf reinen HTML-Seiten ohne Stylesheets und ohne JavaScript haben diese Elemente keine Bedeutung.