

3.6 Linien

horizontale Linien : `<HR>`

3.7 Attribute eines Tags

- Optionen, Parameter
- Betimmen das Wie des Tags

Die Größe

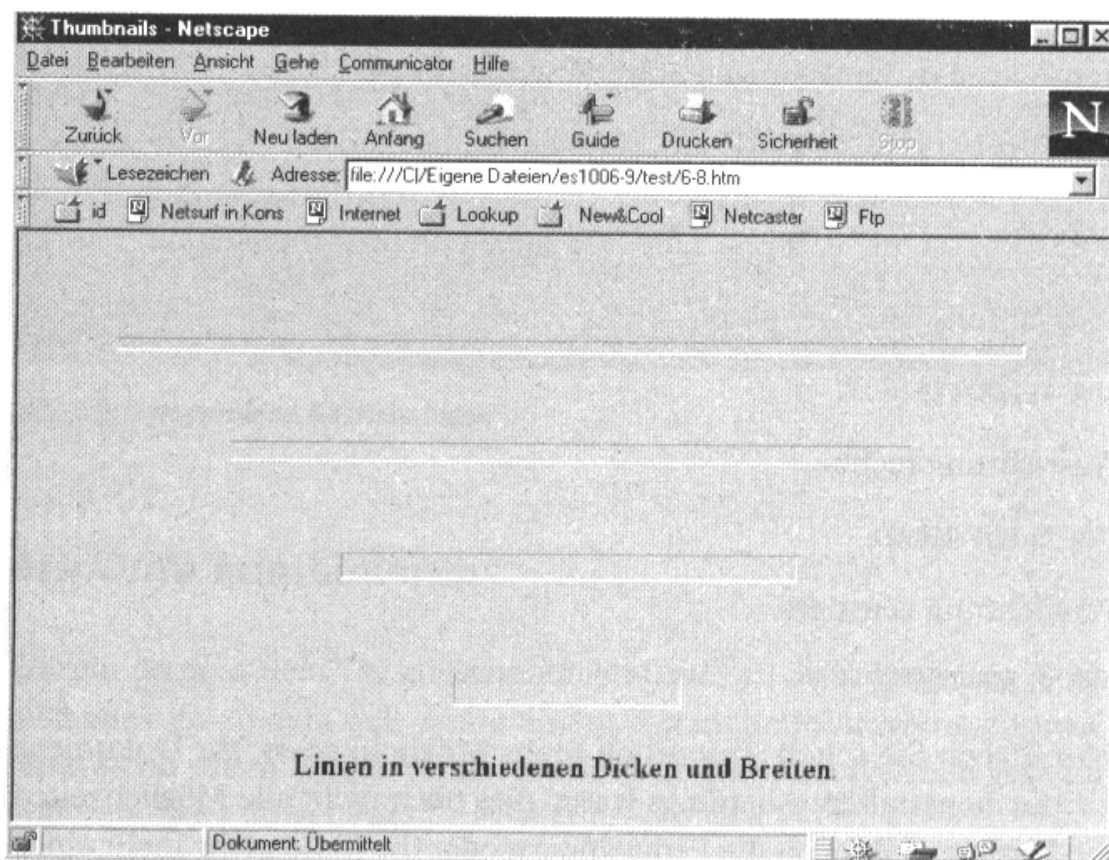
Attribut SIZE

`<HR SIZE=n>`

Die Breite

`<HR WIDE=n>`

`<HR WIDE=n%>`



Linien mit dem Tag `<HR>`

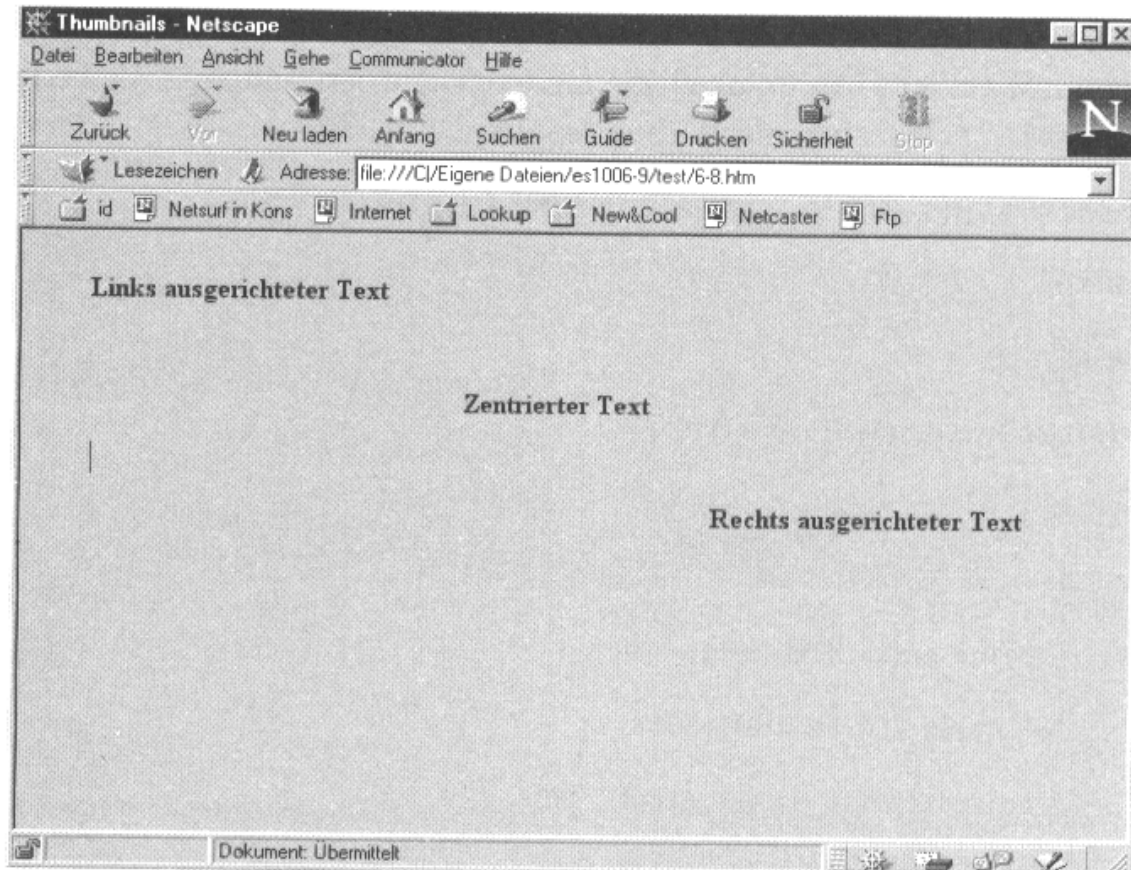
Die Ausrichtung

Attribut ALIGN (engl. ausrichten)

<HR ALIGN=LEFT>

<HR ALIGN=CENTER>

<HR ALIGN=RIGTH>



Verschiedene Ausrichtungen

Attribute kombinieren

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">

<HTML>

<HEAD>

<TITLE> Textausrichtung </TITLE>

</HEAD>

<BODY>

    Meine erste HTML-Seite !

    <HR ALIGN=CENTER WIDTH=50%>

    So schwer ist es gar nicht, <BR>

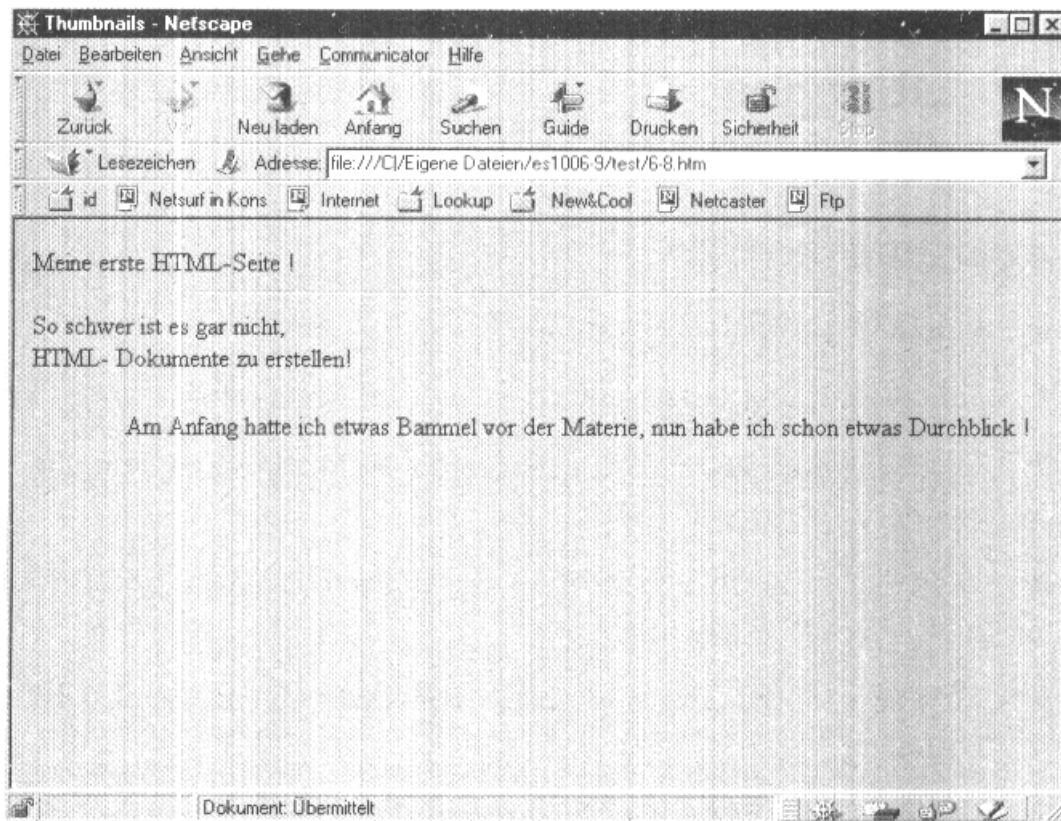
    HTML-Dokumente zu erstellen!

    <P ALIGN=RIGHT>    Am Anfang hatte ich etwas Bammel vor der Materie,

        nun habe ich schon etwas Durchblick ! </P>

</BODY>

</HTML>
```



So setzt der Browser unser Listing dann um

Die *Attribute* werden einfach nacheinander aufgeführt, und sie sind dabei nur durch ein Leerzeichen getrennt:

```
<HR ALIGN=CENTER WIDTH=50%>
```

Die *Attribute* an sich wiederholen sich bei den *Tags*, allerdings sind nicht alle *Attribute* bei allen *Tags* anwendbar.

4 Text und Formatierungen

4.1 Basisformatierungen des Textes

Überschriften

HTML sieht eine Hierarchie von 6 Überschriften vor. Mit dem Tag `<Hn>` beginnt eine Überschrift, und mit dem Tag `</Hn>` wird sie beendet. n wird durch Zahlen zwischen 1 und 6 ersetzt, wobei das Einsetzen der 1 zur größten Überschrift führt. Das nachfolgende Listing verdeutlicht den Einsatz:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">

<HTML>

<HEAD>

<TITLE> &Uuml;berschriften </TITLE>

</HEAD>

<BODY>

<H1> &Uuml;berschrift 1 </H1>

<H2> &Uuml;berschrift 2 </H2>

<H3> &Uuml;berschrift 3 </H3>

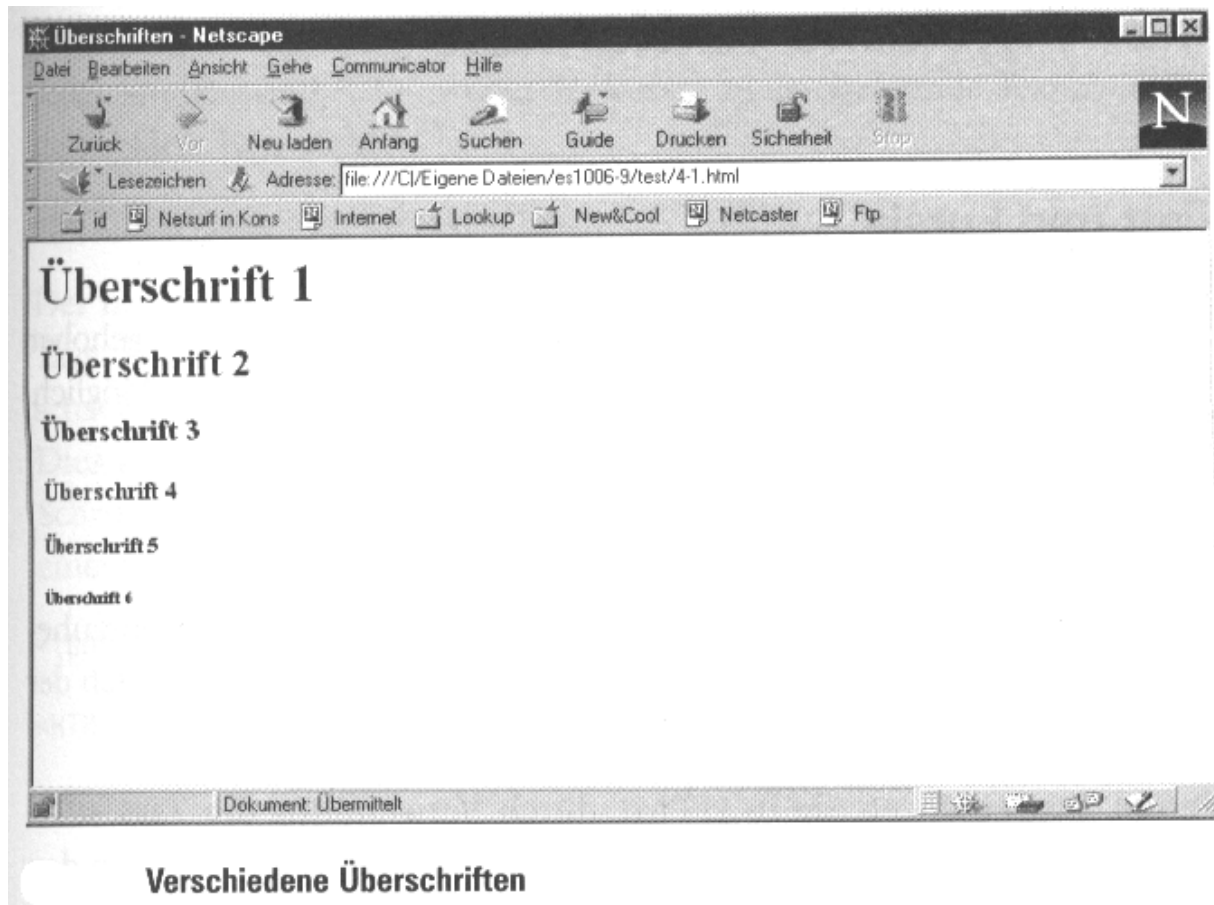
<H4> &Uuml;berschrift 4 </H4>

<H5> &Uuml;berschrift 5 </H5>

<H6> &Uuml;berschrift 6 </H6>

</BODY>

</HTML>
```



4.2 Physische Textformate

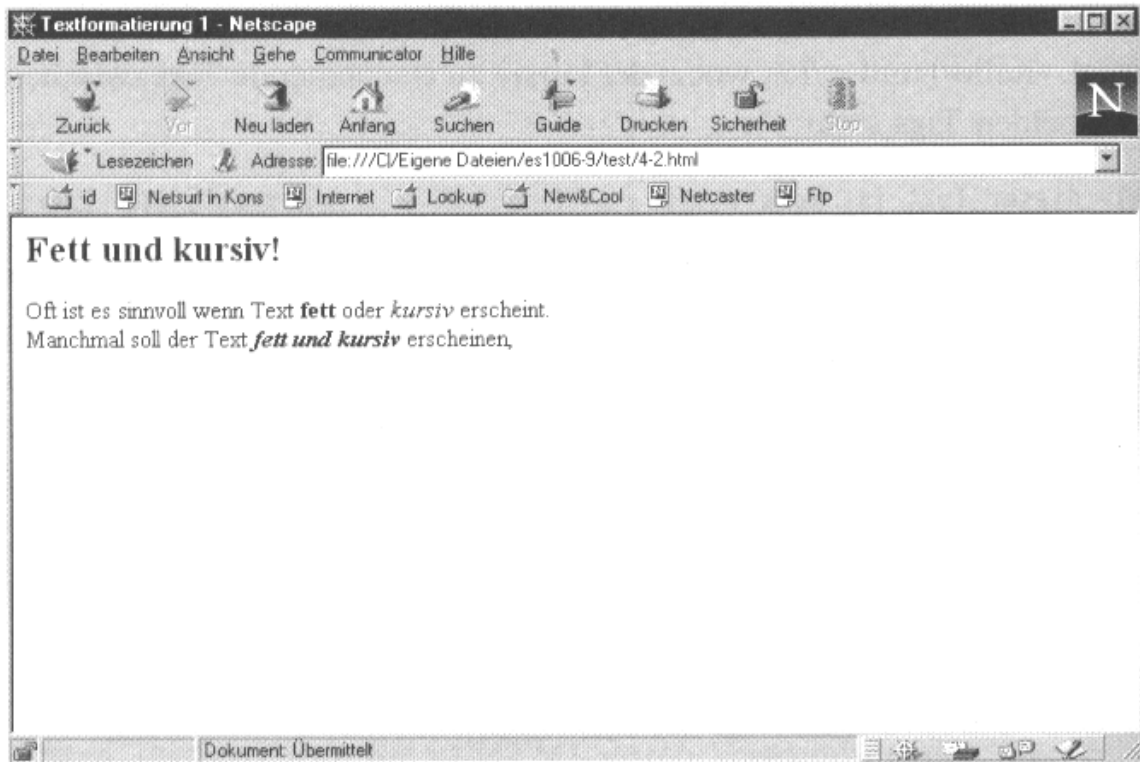
Fett und kursiv

`` (engl. bold)

`<I>` (engl. italik)

Der nachfolgende Quelltext ist ein Beispiel für die Verwendung dieser Tags. Wie Sie in diesem Beispiel sehen, lassen sich diese Tags auch verschachteln, damit Sie z. B. kursiven Text auch noch fett setzen können. Dies geht nicht nur bei diesen beiden Tags, sondern bei allen Tags, die Schriftformatierungen vornehmen. Das Ergebnis, das der Quelltext bei einem Browser hervorruft, sehen Sie in Abbildung 4.2.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD>
<TITLE> Textformatierung 1 </TITLE>
</HEAD>
<BODY>
<H2> Fett und kursiv! </H2>
Oft ist es sinnvoll, wenn Text <B>fett </B>oder <I>kursiv </I>erscheint. <BR>
Manchmal soll der Text <B><I>fett und kursiv</I></B> erscheinen, <BR>
</BODY>
</HTML>
```



Text fett und kursiv

Hoch- und tiefgestellt, groß und klein

^{hochgestellt}

_{tiefgestellt}

<BIG>dieser Text ist größer</BIG>

<SMALL>dieser Text ist kleiner</SMALL>

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Textformatierung 1 </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H2> Großer und kleiner Text, </H2>
```

```
<H2> hoch- und tiefgestellter Text ! </H2>
```

Oft ist es sinnvoll, wenn Text `<BIG> groß </BIG>` oder `<SMALL> klein </SMALL>` erscheint. `
`

Auch läßt sich Text `^{hochstellen}` oder `_{tiefstellen}`. `
`

meine Wohnung hat 50 m `²` `
`

sieht nicht so gut aus, wie `
`

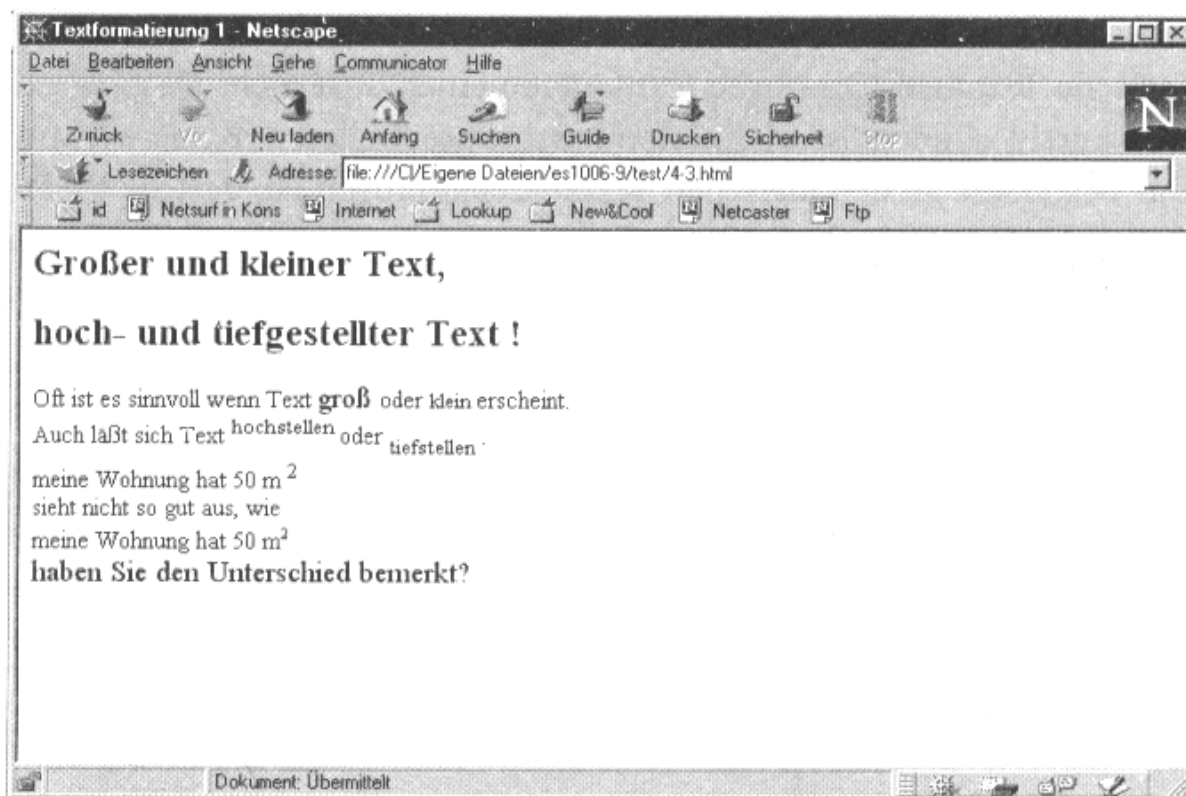
meine Wohnung hat 50 m`^{<SMALL>2</SMALL>}` `
`

`<BIG>` haben Sie den Unterschied bemerkt? `</BIG>` `
`

`</BODY>`

`</HTML>`

Das Ergebnis dieses Quelltextes sehen Sie in Abbildung 4.3.



Text groß, klein, hoch- und tiefgestellt

Nichtproportionale Schrift

`<TT>Schreibmaschinenschrift</TT>`

4.3 Logische Textformate

Logische Textauszeichnungen formatieren einen Text nicht als z. B. fett oder kursiv, sondern sie definieren bestimmte Stellen z. B. als wichtig. Das Besondere an den logischen Textauszeichnungen ist, daß Sie als Ersteller der HTML-Dokumente nicht wissen, wie diese Formatierungen beim Betrachter aussehen werden. Das einzige, das Sie wissen (und das auch nicht hundertprozentig), ist, daß die markierten Textstellen irgendwie verändert sind.

Dies wirkt nur scheinbar unsinnig, denn der Betrachter hat in seinem Browser eingestellt, wie er z. B. eine als Zitat formatierte Textstelle dargestellt haben möchte. Der Vorteil liegt jedoch auf der Hand: Da der Betrachter selber bestimmt, wie solchermaßen markierte Texte dargestellt werden, erkennt er diese auf einen Blick.

Für kommerzielle Werbeseiten im WWW, durchgestylt bis ins Detail, bis zum letzten I-Tüpfelchen auf *Corporate Identity* getrimmt, sind diese Tags sicher nicht sonderlich sinnvoll.

Doch der Ursprung des Web ist die Wissenschaft, und der Anteil an Informationen, wissenschaftlicher und sonstiger Art, bei denen es nicht auf besonders schönes, sondern sinnvolles und übersichtliches Design ankommt, ist immer noch sehr hoch.

Abkürzungen

`<ACRONYM> HTML </ACRONYM>`

Zitate

`<CITE> Dies ist ein Zitat. </CITE>`

Programmlistings

`<CODE> Dies ist ein Listing </CODE>`

Definitionen markieren

`<DFN> Dies ist eine Definition </DFN>`

Nachdrücklich betonter Text (Emphasis)

` das ist mit Nachdruck gesprochen `

Tastatureingaben

`<KBD> Enter </KBD>`

Beispiel

<SAMP> Dies ist ein Beispiel </SAMP>

Wichtiger Text

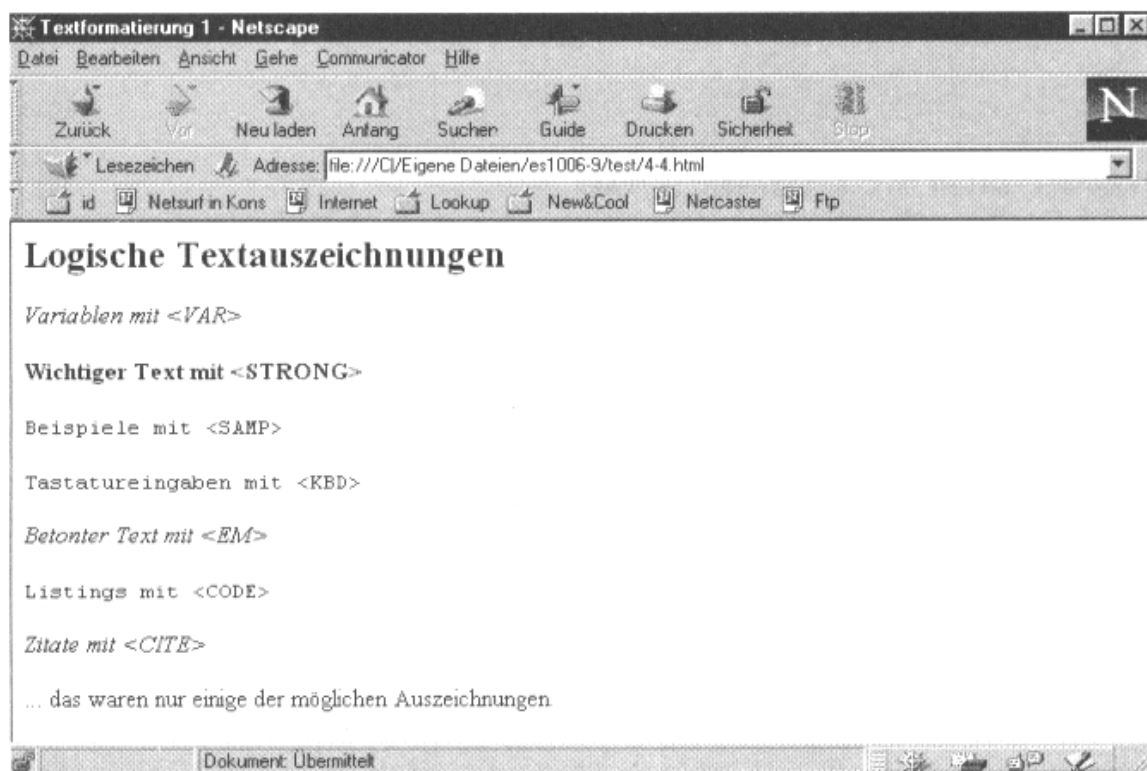
 Dies ist wichtig!

Variablen

<VAR> Variable n </VAR>

Das Erscheinungsbild logischer Textformate

Auch wenn nicht klar ist, wie der Betrachter die logischen Auszeichnungen zu sehen bekommt, ist die Wahrscheinlichkeit sehr hoch, daß er es zu Gesicht bekommt, wie in Abbildung 4.4 zu sehen. Für jedes Tag gibt es Standarddarstellungen, die die meisten Browser zeigen, wenn der Nutzer nichts anderes eingestellt hat.



Logische Textauszeichnungen

4.4 Listen

Listen werden in jedem Textdokument immer wieder benötigt. Da Sie in HTML nicht die Möglichkeiten wie in einer Textverarbeitung haben, sogenannte Tabstops zu definieren, gibt es zum Erstellen von Listen spezielle Tags. Dabei wird zwischen sortierten und unsortierten Listen unterschieden.

Die sortierte Liste

Wenden wir uns zunächst der sortierten Liste zu. In der sortierten Liste werden die Listeneinträge durchnummeriert. Dabei kann die Art der Nummerierung festgelegt werden. Möglich sind arabische und römische Ziffern und alphanumerische Auflistungen.

Eine sortierte Liste wird mit Hilfe des Tags `` definiert. Dabei wird zwischen den Tags `` (engl. »ordered list«) und ``, die die Liste öffnen und schließen, ein weiteres Tagpaar benötigt.

Es sind die Tags `` und ``, zwischen denen die einzelnen Einträge der Liste jeweils aufgeführt werden:

```
<OL>
```

```
<LI> Eintrag </LI>
```

```
</OL>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD>
<TITLE> sortierte Listen </TITLE>
</HEAD>
<BODY>
<H2> verschachtelte Liste </H2>
<OL>
<LI> Liste 1, Eintrag 1 </LI>
<OL>
<LI> Liste 2, Eintrag 1 </LI>
<LI> Liste 2, Eintrag 2 </LI>
</OL>
<LI> Liste 1, Eintrag 2 </LI>
<LI> Liste 1, Eintrag 3 </LI>
</OL>
</BODY>
</HTML>
```

Das Ergebnis zeigt Ihnen die Abbildung 4.6. Wie Sie sehen, ist es etwas unschön, daß die Unterpunkte wieder einfach durchnummeriert sind und nicht mit 1.1, 1.2 usw.

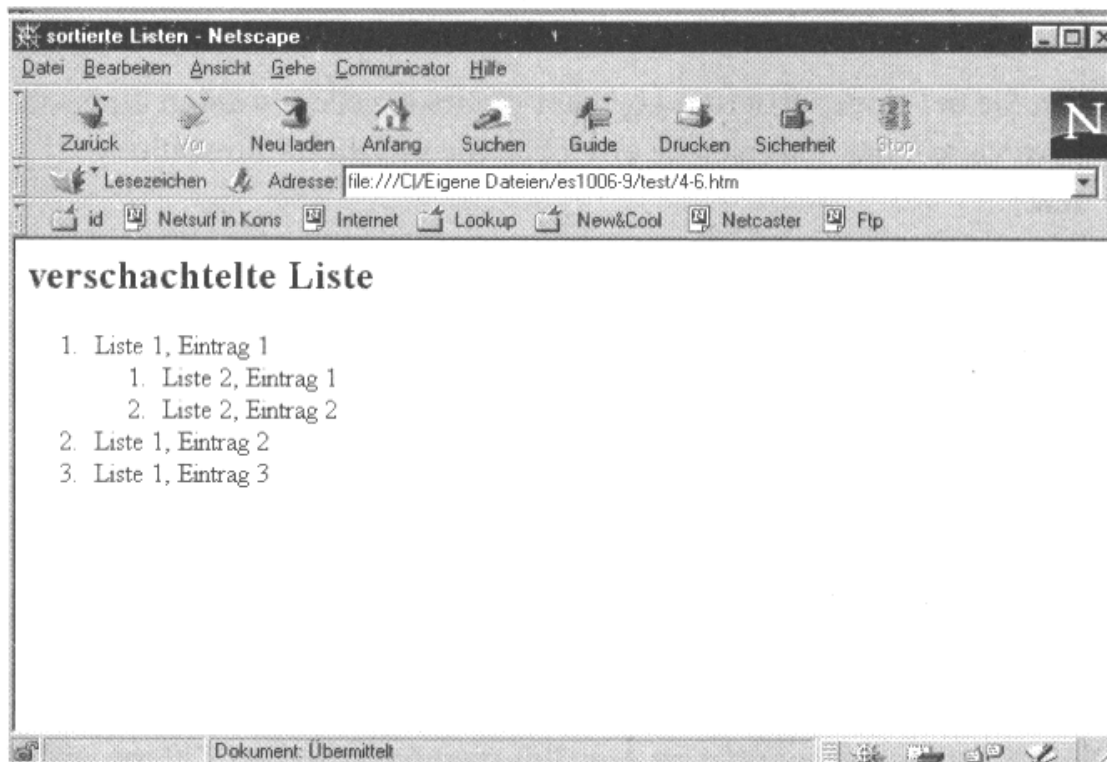


Abb. 4.6: Eine verschachtelte Liste

Abhilfe schafft hier das Attribut `TYPE`, das Sie sowohl bei dem Tag `` als auch bei dem Tag `` einsetzen können. Mit Hilfe dieses Attributs können Sie die oben bereits erwähnte Art der Aufzählung festlegen:

```
<OL TYPE=n> </OL>
```

oder

```
<LI TYPE=n> </LI>
```

Dabei wird *n* durch vorgegebene Werte ersetzt, die jeweils für eine andere Art der Aufzählung stehen:

TYPE=1

Wenn Sie bei einer sortierten Liste gar nichts angeben, werden arabische Ziffern verwendet. Das gleiche erreichen Sie mit dieser Einstellung (1, 2, 3 usw.).

TYPE=a

Diese Einstellung bewirkt eine Aufzählung mit Kleinbuchstaben (a, b, c usw.).

TYPE=A

Großbuchstaben werden hier zur Aufzählung eingesetzt (A, B, C usw.).

TYPE=i

Diese Einstellung bewirkt eine Aufzählung mit römischen Ziffern, wobei hier die Kleinbuchstaben zur Darstellung verwendet werden (i, ii, iii usw.).

TYPE=I

Dies bewirkt die Darstellung römischer Ziffern unter Verwendung großer Buchstaben (I, II, III usw.).

Wenn wir dieses Wissen nun in den vorherigen Quelltext integrieren, sieht die Liste schon viel besser aus. In unserem Beispiel verwenden wir für die Unterlisten Kleinbuchstaben als Numerierung:


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD>
<TITLE> sortierte Listen </TITLE>
</HEAD>
<BODY>
<H2> verschachtelte Liste </H2>
<OL TYPE=1>
<LI> Liste 1, Eintrag 1 </LI>
<OL TYPE=a>
<LI> Liste 2, Eintrag 1 </LI>
<LI> Liste 2, Eintrag 2 </LI>
</OL>
<LI> Liste 1, Eintrag 2 </LI>
<LI> Liste 1, Eintrag 3 </LI>
</OL>
</BODY>
</HTML>
```

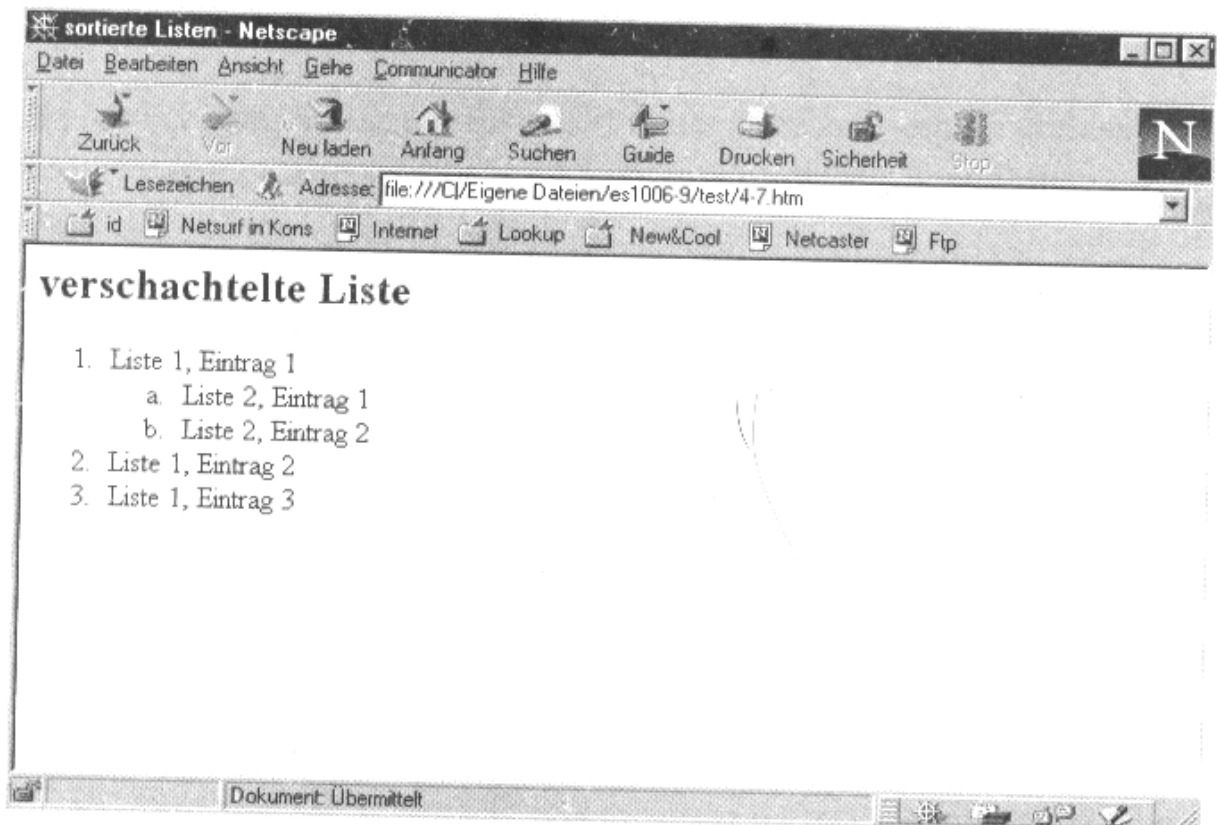


Abb. 4.7: Eine sortierte Liste mit dem Attribut TYPE

Die unsortierte Liste

Nicht immer ist es gewünscht, daß eine Liste durchnummeriert ist. Dazu können Sie unsortierte Listen erstellen, die lediglich mit Aufzählungszeichen verbunden sind, anstatt durchnummeriert zu sein.

Eine unsortierte Liste wird mit Hilfe des Tags `` definiert. Dabei wird zwischen den Tags `` und ``, die die Liste öffnen und schließen, genauso wie bei der sortierten Liste, wieder das Tagpaar `` und `` benötigt:

```
<UL>
```

```
<LI> Eintrag </LI>
```

```
</UL>
```

Sie sehen, der Einsatz erfolgt in der gleichen Art und Weise wie beim Tag ``. Sie können die Beispiele, die wir bei den sortierten Listen bearbeitet haben, nur durch Austausch der Tags `` gegen `` einmal ausprobieren. An der Syntax des Quelltextes ändert sich nichts.

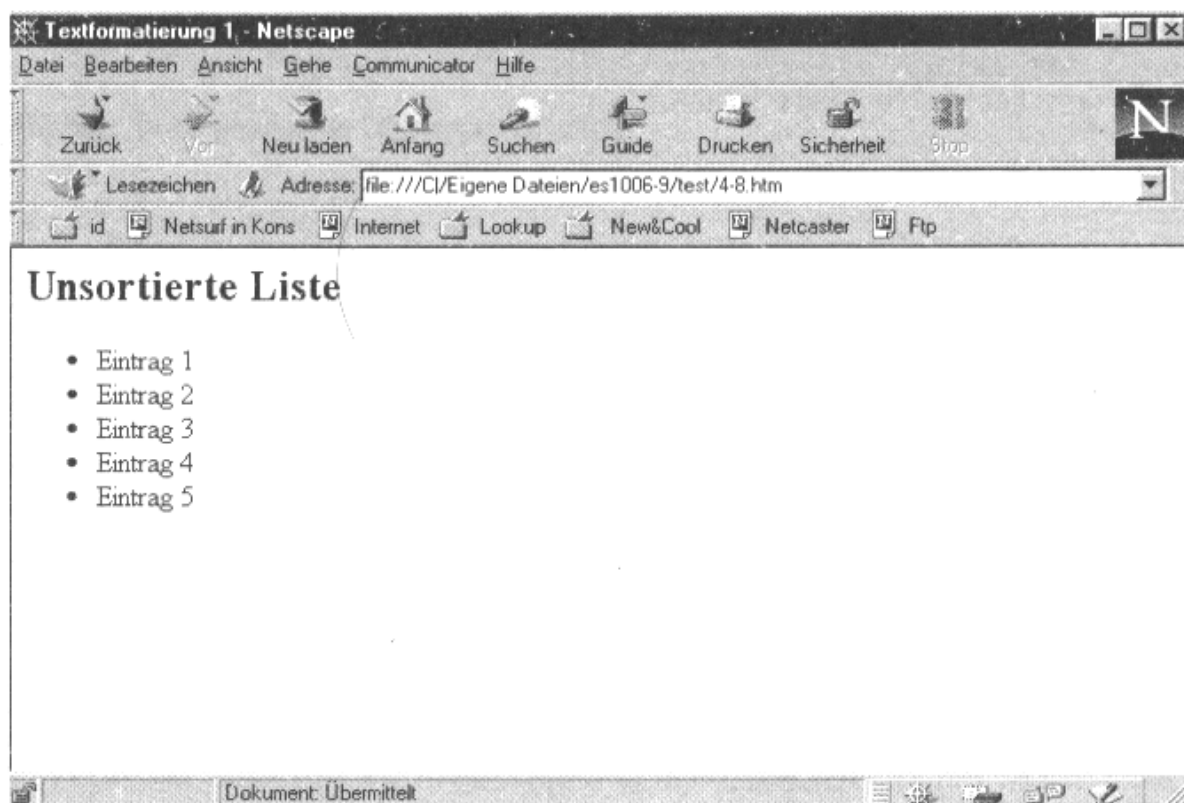


Abb. 4.8: Eine unsortierte Liste

Sortierte und unsortierte Listen verbinden

Weitere interessante Möglichkeiten ergeben sich auch durch die Kombination von sortierten und unsortierten Listen. Das geht ganz einfach. Dazu bearbeiten wir das folgende Beispiel, das im Browser ein Bild wie in Abbildung 4.9 hervorruft.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">

<HTML>

<HEAD>

<TITLE> Listen </TITLE>

</HEAD>

<BODY>

<H2> verschachtelte Liste </H2>

<OL>

<LI> sortierte Liste 1, Eintrag 1 </LI>

<UL>

<LI> unsortierte Liste 2, Eintrag 1 </LI>

<LI> unsortierte Liste 2, Eintrag 2 </LI>

</UL>

<LI> sortierte Liste 1, Eintrag 2 </LI>

<LI> sortierte Liste 1, Eintrag 3 </LI>

</OL>

</BODY>

</HTML>
```

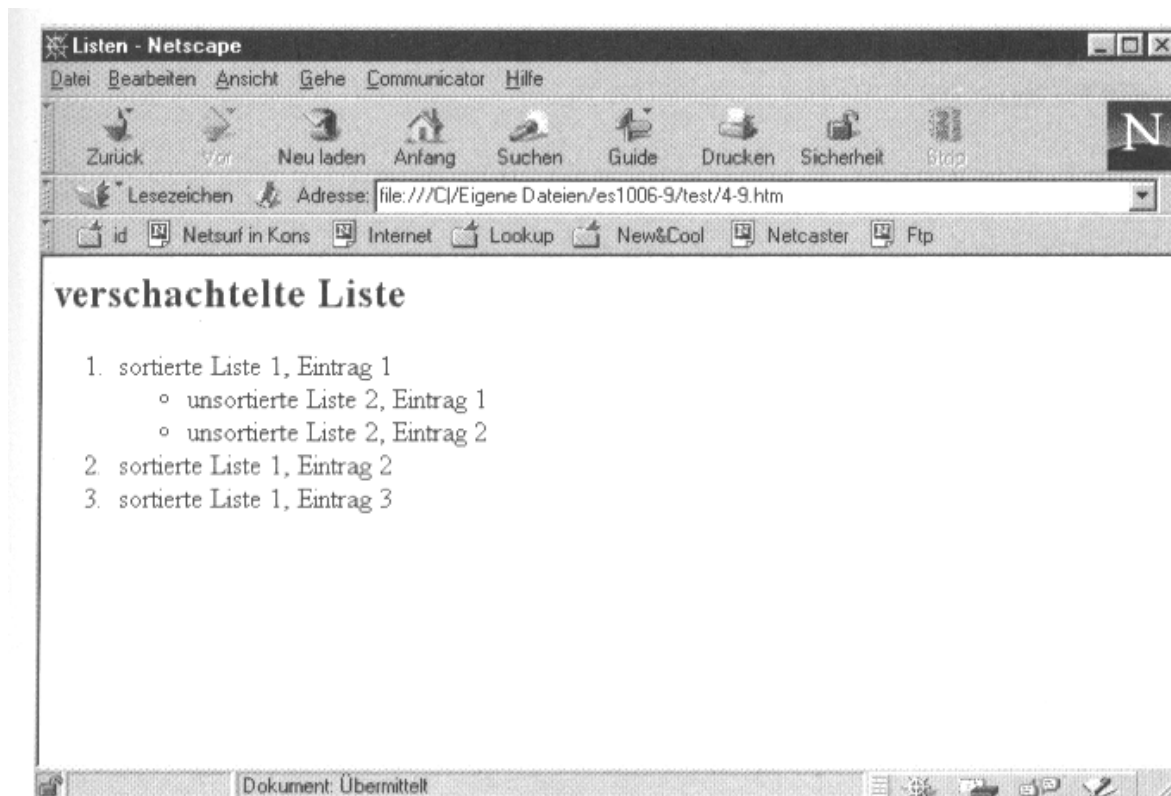


Abb. 4.9: Auch gemischte Listen sind möglich

4.5 Glossar

Ein Glossar ist ein Fachbegriffverzeichnis, das meist am Ende eines Buches steht. Gerade bei wissenschaftlichen Texten wird oft ein Glossar benötigt. Wenn HTML dafür keine Funktion hätte, würde ich nichts über Glossare schreiben; kommen wir also zu den benötigten Tags.

Der Syntaxaufbau eines Glossars in HTML ist dem einer Liste gleich, lediglich die zu verwendenden Tags sind andere.

Ein Glossar wird mit Hilfe des Tags `<DL>` definiert. Dabei werden zwischen den Tags `<DL>` und `</DL>`, die die Liste öffnen und schließen, weitere Tags benötigt. Es sind die Tags `<DT>` und `</DT>`, zwischen denen die einzelnen Begriffe des Glossars aufgeführt werden.

Die Erklärung des Begriffs wiederum wird zwischen die Tags `<DD>` und `</DD>` geschrieben:

In die Praxis umgesetzt ist dies im folgenden Beispiel:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN//">
<HTML>
<HEAD>
<TITLE> Glossar </TITLE>
</HEAD>
<BODY>
<H2> Glossar </H2>
<DL>
<DT> HTML 4 </DT>
<DD> HTML ist die Programmiersprache des World Wide Web. Die Version 4
ist dabei die derzeit gültige Fassung. Einige Tags wurden erweitert
oder ganz neu eingeführt, andere fielen ganz weg. </DD>
<DT> Java </DT>
<DD> Java ist ein bla bla bla bla..... , Ihnen ist jetzt sicher klar,
wie es funktioniert. </DD>
</DL>
</BODY>
</HTML>
```

Abbildung 4.10 zeigt Ihnen das Glossar aus obigem Quelltext.

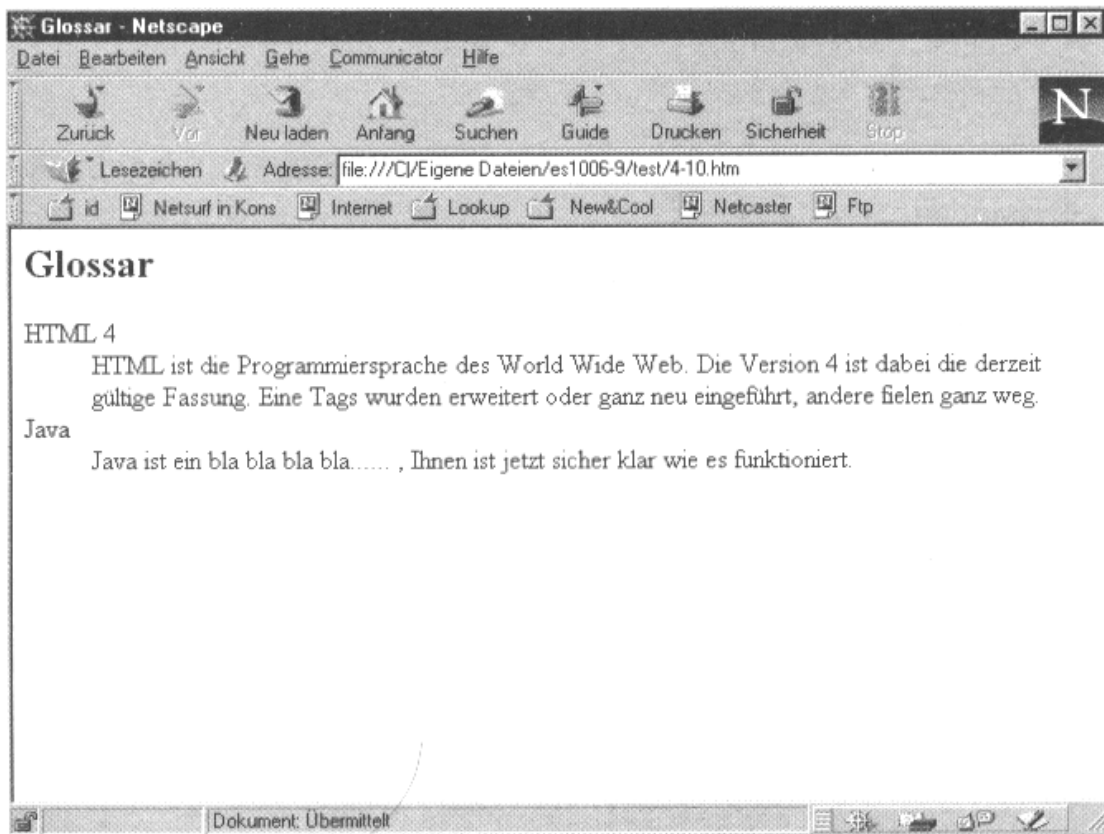


Abb. 4.10: Glossar in HTML