

Publizieren im Internet

HTML

&



Dipl. Ing. Larissa Naber

Mission Impossible?

Ihr Auftrag: Bringen Sie die Firma Noriginal Art ins Internet!

Die Firma Noriginal Art fertigt und vertreibt hochwertige Imitationen bekannter Kunstwerke der klassischen Moderne. Die Firma benötigt einen kleinen Webauftritt mit einer Preisliste, Künstlerbiographien, einem Glossar, einer kleinen Galerie und einer Kontaktmöglichkeit.

Folgende Seiten sollen generiert werden:

<code>index.html</code>	Startseite
<code>stile.html</code>	Glossar
<code>kuenstler.html</code>	Künstlerbiographien
<code>preise.html</code>	Preisliste
<code>werke.html</code>	Galerie
<code>kontakt.html</code>	Kontaktformular

Inhaltsverzeichnis

1	HTML ist keine Hexerei!	1
1.1	Die erste Seite	1
1.2	Textfluss & Absatzkontrolle	2
1.3	Hyperlinks einbinden	4
1.4	Ein Bild sagt mehr als 1000 Worte	7
1.4.1	Bilder einfügen	7
1.4.2	Bilder verlinken	7
1.4.3	Tricks mit Bildern	10
1.5	Listen	11
1.5.1	Glossare - Definitionslisten	12
1.6	Sonderzeichen – HTML Entities	13
1.7	Textauszeichnung	14
2	Jetzt wird es bunt!	17
2.1	HTML und Farbe	17
2.1.1	Browsersichere Farben	18
2.1.2	Globale Farbeinstellungen	18
2.2	Textgestaltung	19
3	Tabellen	20
3.1	Daten in Reih und Glied	20
3.2	Tabellen formatieren	22
3.2.1	Größenangaben	22
3.2.2	Ausrichtung	23
3.2.3	Farben	25
3.3	Tabellen zum Layouten	26
3.3.1	Tabellen verschachteln	26
3.3.2	Fixe oder flexible Layouts?	27
3.3.3	Farbflächen und der Seitenrand	28
3.3.4	Zeilenhöhe und unerwünschte Freiräume	30
3.3.5	Graphiken und das colspan Attribut	31
3.4	Tabellen für Graphikmosaike	32

4	Formulare	34
4.1	Grundlagen	34
4.2	CGI	35
4.3	Formularfelder	35
4.3.1	Texteingabefelder	35
4.3.2	Auswahlmöglichkeiten	38
4.3.3	Buttons	41
4.4	Weitere Formularfelder	42
5	Meta-Informationen	44
5.1	Infos für Suchmaschinen	44
5.2	Infos für den Browser	45
6	Frames	46
6.1	Funktionsweise	47
6.2	Formateinstellungen	48
6.3	Framesets verschachteln	49
6.4	Framesets und Hyperlinks	50
6.4.1	Attribute	51
7	CSS - Cascading StyleSheets	52
7.1	Das erste Stylesheet	52
7.2	CSS Grundlagen	53
7.3	Funktionsweise	54
7.3.1	CSS Syntax	54
7.4	Verknüpfung	55
7.5	Vererbung & Kaskade	56
7.6	Selektoren	57
7.6.1	Class Selektor	58
7.6.2	Kontextuelle Selektoren	59
7.7	Maßeinheiten	60
7.7.1	Größenangaben	60
7.7.2	Farbangaben	60
7.7.3	Schriftgrößen	61
7.8	CSS-Regeln	62
7.8.1	Textformate	62
7.8.2	Absatzformate	63
7.8.3	Boxen & Rahmen	63
7.8.4	Hintergrund Eigenschaften	65
	Anhang	67

Abbildungsverzeichnis

1.1	Die erste HTML Seite	1
1.2	Absätze und Zeilenumbrüche	3
1.3	Überschriften und Trennlinien	4
1.4	Hyperlinks	6
1.5	Verweise im Dokument	6
1.6	Bilderpositionierung und Textfluß	10
1.7	Listen und Nummerierungen	12
1.8	Glossarliste	13
1.9	Textauszeichnung	15
3.1	Tabellenbreite und Spaltenbreite	23
3.2	Wirkung des <code>align</code> Attributs	24
3.3	Wirkung des <code>valign</code> Attributs	25
3.4	Leftmargin und topmargin Attribute	29
3.5	Leftmargin und topmargin Attribute	30
3.6	Tabellenzeilenhöhe	31
3.7	Probleme mit dem <code>colspan</code> Attribut	32
4.1	Ein einfaches Formular	34
4.2	Ein einzeliges Texteingabefeld	36
4.3	Passwortfeld	36
4.4	Editorfeld	37
4.5	Checkboxen	38
4.6	Ein Gruppe Radiobuttons: Es kann immer nur ein Ra- diobutton angewählt sein.	39
4.7	Auswahlfelder und Boxen	41
4.8	Fileuploadfeld	43
5.1	Metainformationen	45
6.1	Ein einfaches Frameset	46
6.2	Horizontale Aufteilung mit Frames	48
6.3	Drei Frames	50
7.1	Ein einfaches Stylesheet	53
7.2	Class Selektoren	58

7.3 Kontextuelle Selektoren 60
7.4 Border, Margin & Padding 64

HTML ist keine Hexerei!

1.1 Die erste eigene HTML Seite

```
<html>
  <head></head>
  <body>
    HTML ist ganz einfach!
  </body>
</html>
```

Speichern Sie Datei unter dem Namen `index.html` und betrachten Sie das Ergebnis im Browser. Nach jeder Änderung im Quellcode müssen Sie die Datei speichern und im Browser neu laden.

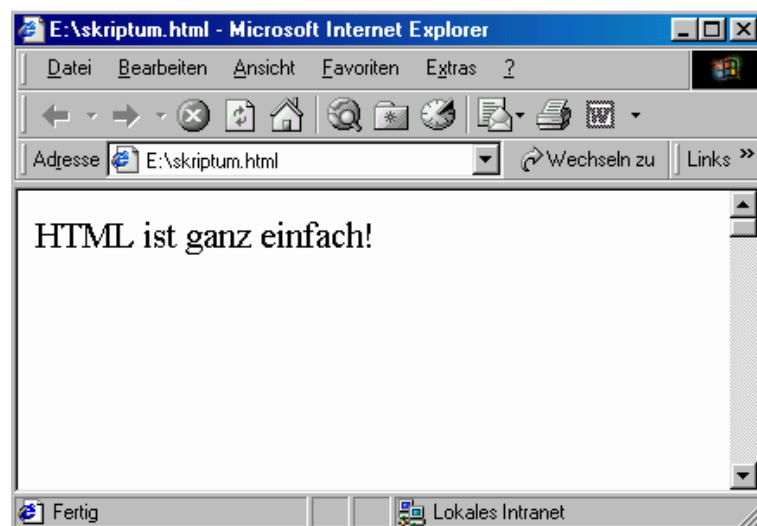


Abbildung 1.1: Die erste eigene HTML Seite! Das ist das Ergebnis des obigen Quellcodes.

Die Texte in spitzen Klammern heißen Tags. Tags treten immer paarweise auf. Das schließende Tag trägt den selben Namen wie das Öffnende, aber mit einem Schrägstrich davor: `<html> ... </html>`. Das Paar `<html> ... </html>` beinhaltet die gesamte **HTML** Seite. `<body>` und `</body>` umschließen den eigentlich Seiteninhalt (den Textkörper, **body**): Alles was enthalten ist, wird vom Browser im Fenster dargestellt. Zwischen `<head>` und `</head>` können quasi im „Seitenkopf“ (**header**) zusätzliche Informationen über die Seite untergebracht werden, die wir aber derzeit noch nicht benötigen (siehe Kapitel 5). Einstweilen können Sie `<head>` und `</head>` weglassen.

`<html>``<body>``<head>`

In den Tags wird Groß und Kleinschreibung nicht unterschieden: `<html>` = `<HTML>` = `<hTML>`. Der designierte Nachfolger zu HTML 4.0 - XHTML 1.0 - ist allerdings schon spezifiziert und basiert auf der Sprache XML, die wesentlich strengere Maßstäbe anlegt: In XHTML müssen alle Tags klein geschrieben werden und alle Endtags müssen ausnahmslos vorhanden sein.

XHTML nimmt alles viel genauer als HTML

Vorsicht Falle!

Die häufigste Fehlerquelle sind vergessene (oder verkehrte) spitze Klammern!



1.2 Textfluss & Absatzkontrolle

Schreiben Sie jetzt einen längeren Text in die HTML Datei. Sie sehen, dass der Browser den Text an die Fenstergröße anpasst: er betrachtet Ihre Eingabe als Fließtext. Ändern Sie die Größe des Browserfensters, so wird der Browser den Text neu umbrechen. Das ermöglicht es, den Inhalt einer HTML Seite auch auf Geräten mit kleinen Bildschirmen zu betrachten. Leider hat es den Nachteil, dass der Browser neue Zeilen, Absätze und zusätzliche Leerzeichen als Gestaltungsmittel ignoriert.

Um Absätze zu generieren umschließen Sie den Text, der den Absatz (**paragraph**) bilden soll, mit `<p>` und `</p>`. Möchten Sie nur eine neue Zeile ohne einen eigenen Absatz (kein zusätzlicher Abstand zwischen den Zeilen) schreiben sie `
` (**break**)(in diesem Fall dürfen Sie das schließende Tag `</br>`) weglassen, in XHTML schreiben Sie: `
`.

`<p>``
`

```
<html>
  <head></head>
  <body>
    <p>Das ist Text in einem eigenen Absatz.</p>
    <p>Ein weiterer Absatz, diesmal mit<br>
      einer neuen Zeile.
```

```

</p>
</body> </html>

```



Abbildung 1.2: Eine einfache Seite mit Absätzen (<p>) und einem Zeilenumbruch (
)

Vorsicht Falle!

Beim Formatieren von Absätzen fügt der Browser den zusätzlichen Zeilenabstand an der Stelle des öffnenden <p> Tags ein!



Nachdem wir die Kontrolle über die Absätze wieder erlangt haben, möchten wir den Text mit Überschriften (heading) weiter gliedern. HTML stellt uns sechs verschiedene Überschriften zur Verfügung: <h1> ..</h1>, <h2>..</h2>, ..., <h6>..</h6>. Die Nummer entspricht jeweils der Verschachtelungstiefe. <h1> ist eine Überschrift der Ebene 1 und kann alle weiteren enthalten. Dementsprechend nimmt die Schriftgröße von <h1> bis <h6> ab.

Um Abschnitte optisch stärker zu trennen, benutzen Sie das <hr> Tag für horizontale Linien (horizontal rulers). Auch beim <hr> Tag können Sie das Ende Tag weglassen (in XHTML schreiben Sie wieder <hr />.

```

<html>
  <head></head>
  <body>
    <h1>HTML Einführung</h1>
    <h2>HTML ist ganz einfach!</h2>

```

```

        HTML betrachtet jeden Text als Fließtext
        <hr>
        Autor: HTML Kurs
    </body>
</html>

```

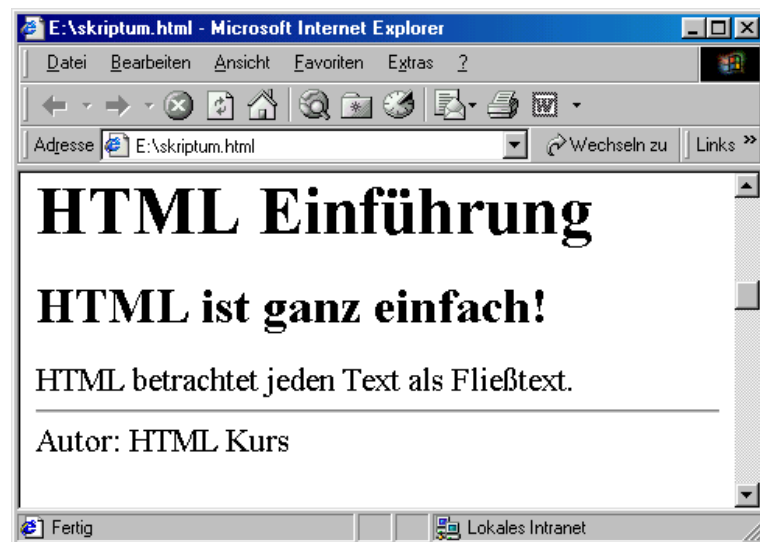


Abbildung 1.3: Eine einfache Seite mit Überschriften (`<h1>`,`<h2>`) und einer Trennlinie (`<hr>`)

Übung:

Öffnen Sie die Datei index.html (Normaler Art Startseite) und benutzen Sie die uns bereits bekannten Formatierungen. Details auf Übungsbogen 1



1.3 Was wäre das WWW ohne Hyperlinks?

Referenzen ([hyperreference](#)) auf andere Seiten (links) werden im Text mit dem `<a>`Tag ([anchor](#)) verankert

```
<a href="andereseite.html">eine andere lokale Seite</a>
```

linkt auf eine andere Seite im gleichen Verzeichnis,

```
<a href="http://www.xyz.at/ab.html"> eine fremde Seite</a>
```

linkt auf eine Seite auf einem anderen Webserver. Wenn eine lokale Seite nicht im gleichen Verzeichnis gelegen ist, muss auch der Pfad angegeben

`<a>`

werden:

```
<a href="produkte/computer/laptops.html">Laptops</a>
```

Der Text `href` bezeichnet ein sogenanntes Attribut. Attribute enthalten Zusatzinformationen zu den Tags. Die allgemeine Form lautet: `<tagname attributname="attributwert">test</tagname>`. Der Attributwert sollte immer von Doppelhochkommas umschlossen sein.

Vorsicht Falle!

Vergessen Sie nicht die Doppelhochkommas wieder zu schließen!



Ein Tag kann auch mehrere Attribute haben, diese werden einfach hintereinander gestellt. Die Reihenfolge der Attribute ist unerheblich.

```
<tagname att1="att1wert" att2="att2wert"
      attz="atzwert">
  text
</tagname>
```

```
<html>
  <head></head>
  <body>
    <h1>HTML Einführung</h1>
    Das
    <a href="http://www.w3c.org">
      World Wide Web Consotium (W3C)
    </a>
    gibt die HTML Spezifikation heraus.
  </body>
</html>
```

Übung:

Setzen Sie die Links in der Noriginal Startseite (`index.html`). Details dazu in Übungsbogen 2.



Man kann auch Links innerhalb eines (langen) Dokuments setzen: Der Text, der als Ziel des internen Hyperlinks dienen soll, erhält einen Namen (`name`): `1: `. Ein Hyperlink in der Seite verweist auf diese Textstelle: `[1]`. Das Zeichen `#` kommt nur im Verweis (`<a href ...>`) vor.



Abbildung 1.4: Ein Hyperlink auf die Webseite des W3C.

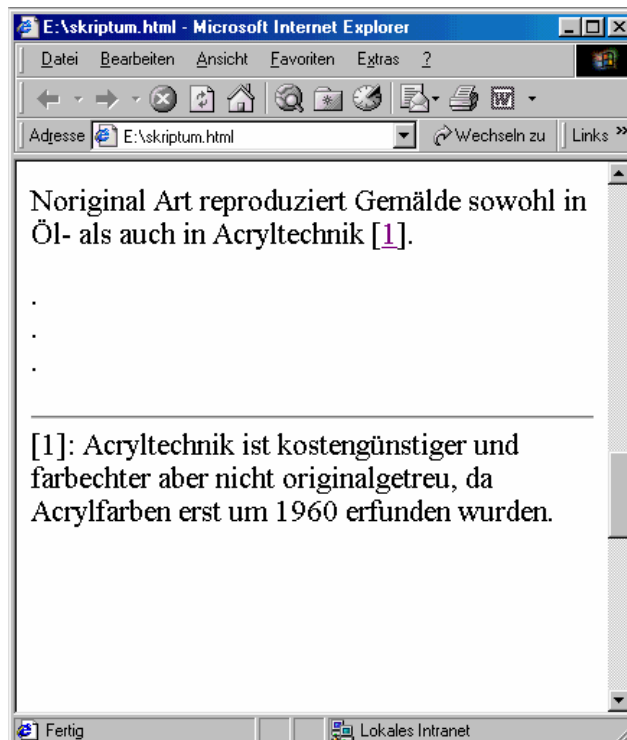


Abbildung 1.5: Mit dem <a> Tag kann man auch Verweise innerhalb des Dokuments setzen. Das Beispiel zeigt eine typische Anwendung als Fußnote innerhalb der gleichen Seite.

1.4 Ein Bild sagt mehr als 1000 Worte

Die ersten Webseiten kamen ganz ohne Bilder aus - heute sind Bilder aus Webseiten fast nicht mehr wegzudenken. Am gebräuchlichsten (und am besten unterstützt) sind GIF und JPEG (Joint Photographic Expert Group) Dateien. Neuere Browser unterstützen auch das PNG Format.

Bilder einfügen

Eine Graphik (`image source`) binden Sie mit `` ein (auch hier hat es sich eingebürgert das `` Tag wegzulassen; XHTML: ``) Das `` Tag unterstützt mehrere Attribute. Eines der wichtigsten ist das `alt` Attribut: `` mit dem Sie eine Bildlegende (`alternate text`) speichern können. Der Text erscheint, wenn die Graphik noch nicht geladen ist (oder der Browser keine Graphiken darstellen kann) und er erscheint auch als Tooltipp, wenn die Maus länger über einer Graphik verweilt.

Vorsicht Falle!

Das Attribut, das den Namen des Bildes angibt heißt `src` nicht `scr` (Sport Club Rapid)!



Übung:

Fügen Sie das Bild `noriginal.jpg` auf der Startseite (`index.html`) anstelle der Überschrift ein.



Ebenfalls sehr wichtig sind das `height` und das `width` Attribut: diese sagen dem Browser wie viel Platz in Breite (`width`) und Höhe (`height`) er für das Bild reservieren muss. Werden diese Angaben nicht gemacht, hält der Browser einen kleinen quadratischen Platz frei und formatiert die Seite neu, sobald er die Graphik vollständig geladen hat und die genauen Abmessungen der Graphik kennt:

``. Eine Graphik kann mit Hilfe dieser Angaben auch skaliert werden.

Bilder verlinken

Bilder werden in Menüleisten häufig als Link eingesetzt:

```
<a href="home.html">  
    
</a>
```

Das so verlinkte Bild erscheint allerdings mit einem blauen Rahmen. Um den (in den meisten Fällen störenden) Rahmen ([border](#)) abzdrehen, benutzen Sie das Attribut `border="0"` im `` Tag:

```

```

Exkurs: Graphikformate im WWW

Das GIF-Format

Das GIF-Format gilt als das „Haus-“ Format von CompuServe und wurde speziell für den Online-Einsatz entwickelt. Es zeichnet sich durch eine hohe Komprimierungsdichte aus. Derzeit aktuell ist das „89er-Format“, dass das Original um Transparenz, Interlacing (Graphik wird nicht zeilen- sondern schichtweise aufgebaut; so erhält man schneller einen Eindruck von der Graphik), und Animation erweitert.

Ein Nachteil des GIF-Formats ist, dass es maximal 256 Farben pro Datei speichern kann; das allerdings verlustfrei. Aufgrund der limitierten Farbzahl eignet sich Gif nicht für Photos, oder feine Farbabstufungen, sondern eher für Clipartähnliche Inhalte (Strichgraphiken, Buttons, Texte in Graphikform). Große einheitliche Farbflächen lassen sich besonders gut komprimieren.

Das JPEG-Format

Bei JPEG, entwickelt von der Joint Photographic Expert Group, komprimiert ebenfalls sehr gut (aber verlustbehaftet) und hat gegenüber dem GIF-Format den Vorteil, dass es pro Bild bis zu 16,7 Millionen Farben speichern kann. Aufgrund seiner Charakteristik eignet sich das JPEG-Format vor allem zum WWW-gerechten Abspeichern von eingescannten Fotos, aber auch für andere Grafiken, in denen sehr feine Farbverläufe vorkommen. Die Qualität der Datei regeln Sie über den Kompressionsfaktor: Je höher der Kompressionsfaktor, desto schlechter die Bildqualität. Ein Komprimierungsfaktor von 5 bis 10 ist für eine gute Bildqualität (bei Weiterverarbeitung) ein sinnvoller Wert, je nach Inhalt (je unruhiger das Bild desto leichter kann man es komprimieren) sind für das Web Kompressionraten bis 70% verträglich. Dem Interlaced Modus beim GIF, entspricht das progressive JPEG - auch hier wird das Bild schichtweise detaillierter.

Das PNG-Format

PNG bedeutet Portable Network Graphic (ausgesprochen: PING). Es handelt sich um ein Grafikformat, das eigens für den Einsatz im WWW konzipiert wurde. PNG soll alle Vorteile von GIF und JPEG in sich vereinen:

PNG komprimiert verlustfrei und unterstützt Transparenz wie das GIF-Format, und unterstützt 16,7 Mio. Farben wie das JPEG-Format.

Übung:

Fügen Sie die Bilder Stile, Werke und Künstler auf der Startseite ein, und verlinken Sie diese auf die zugehörigen Seiten. Details in Übungsbogen 3.

*Tricks mit Bildern*

Normalerweise setzt das Bild auf der Textzeile auf, der Text wird danach fortgesetzt. Wenn Sie wollen, dass der Text um das Bild fließt, benutzen Sie die Attribute `align="left"` oder `align="right"`. Ein `<br clear="all">` hebt die Wirkung von `align` wieder auf, und der Text wird unter der Graphik fortgesetzt.

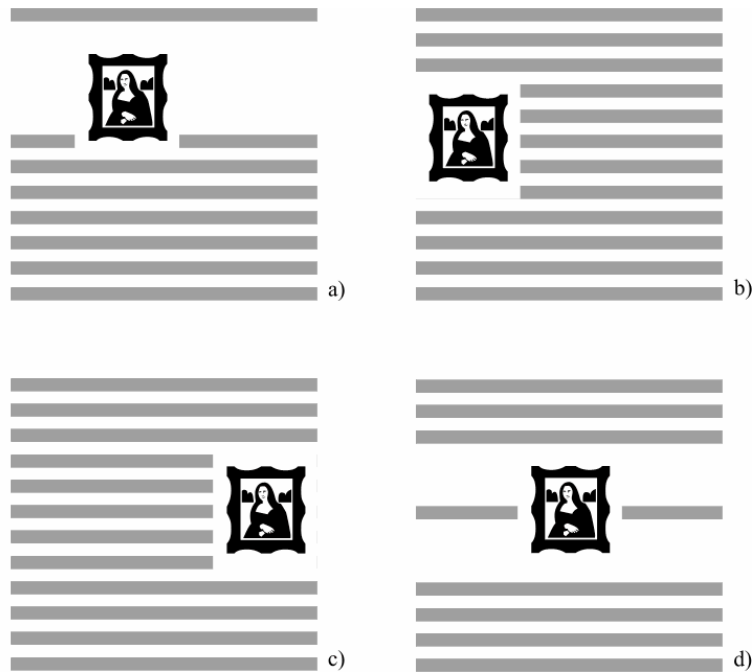


Abbildung 1.6: Mit dem `align` Attribut kann man Bilder auf verschiedene Art und Weise in den Text integrieren. Abbildung a) zeigt die Position ohne `align` Attribut. Abbildung b) verdeutlicht den Effekt von `align="left"`: das Bild rückt nach links, der Text fließt rechts vorbei. Abbildung c) zeigt `align="right"` und Abbildung d) `align="center"`

Übung:

Fügen Sie die Bilder auf der Seite `kuenstler.html` ein. Benutzen Sie `align="left"` und `<br clear="all">` um den gewünschten Effekt zu erreichen (Details in Übungsbogen 4).



1.5 Listen

Um Aufzählungen zu präsentieren oder um Texte prägnanter zugestalten, eignen sich Listen vorzüglich.

HTML bietet zwei verschiedene Sorten: Aufzählungen (`unordered list`) `` und nummerierte Listen (`ordered list`). Die einzelnen Elemente werden `` jeweils mit ` .. ` angegeben (`list item`). ``

```
<ul>
  <li>Äpfel</li>
  <li>Birnen</li>
  <li>Zwetschken</li>
</ul>
```

```
<ol>
  <li>Erstens</li>
  <li>Zweitens</li>
  <li>Drittens</li>
</ol>
```

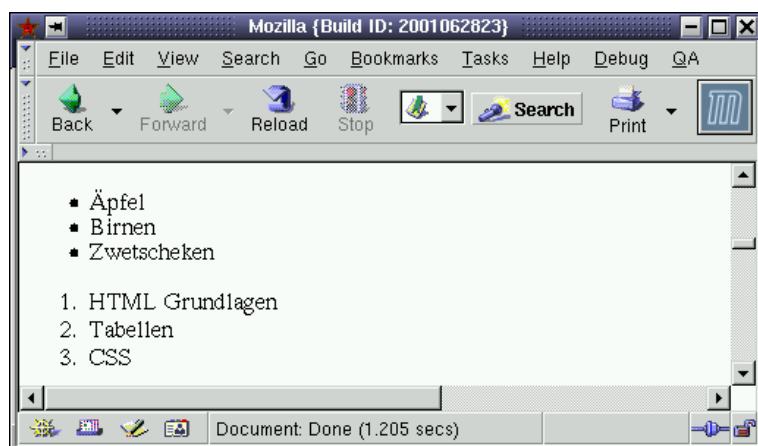


Abbildung 1.7: Die obere Abbildung zeigt eine `` Liste, die untere eine `` Liste.

Übung:

Setzen Sie die Aufzählung auf der Noriginal Startseite (`index.html`) als ``.



Glossare - Definitionslisten

Eine weitere Art Liste ist das Glossar (definition list), das ein lexikonähnliches Layout ermöglicht. Ein Glossar verfügt immer über mehrere Eintragspaare: den Term (definition term) und die eigentliche Definition (definition data) .

`<dl>`
`<dt>`
`<dd>`

```
<dl>
  <dt>WWW</dt>
  <dd>World Wide Web - weltweiter Verbund von
    Informationsseiten</dd>
  <dt>HTML</dt>
  <dd>Sprache in der WWW-Seiten geschrieben
    werden</dd>
  <dt>HTTP</dt>
  <dd>Protokoll das zum Übertragen der
    WWW-Seiten dient</dd>
</dl>
```

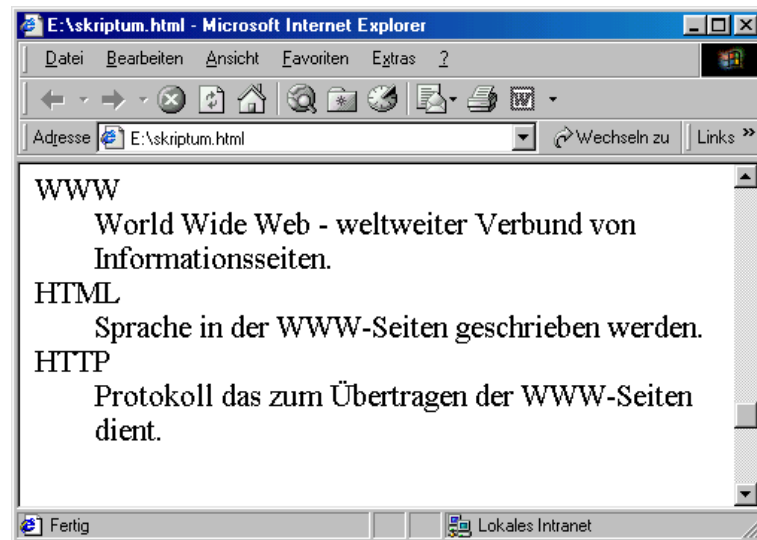


Abbildung 1.8: Glossareinträge lassen sich mit den Tags `<dl>`, `<dt>` und `<dd>` erstellen. Die Umbrüche werden automatisch vom Browser erstellt.

Übung:

Machen Sie ein Glossar für die Begriffe Impressionismus, Expressionismus, Kubismus in der Datei `stile.html`.



1.6 Sonderzeichen – HTML Entities

HTML unterstützt grundsätzlich keine Zeichen außerhalb des ASCII Zeichenraums, auch wenn modernen Browser Umlaute oder das Scharfe s (β) im allgemeinen akzeptieren. Das kann allerdings dann zu Problemen führen, wenn das Zeichen in einem falschen Zeichensatz interpretiert wird. Ein Amerikaner wird ein \ddot{u} , oder β sehen, weil er den latin-1 Zeichensatz benutzt, der auch in Westeuropa gebräuchlich ist, unsere östlichen Nachbarn (latin-2 Zeichensatz für Osteuropa) würden statt dessen andere Zeichen sehen. Deshalb sollte man immer die von HTML vorgesehenen Mechanismen (HTML Entities) zum Erzeugen von Sonderzeichen benutzen. Die für uns wichtigsten sind:

ä = `ä` (a Umlaut)
 ü = `ü`
 ö = `ö`
 Ä = `Ä` (Ö, Ü ditto)
 ß = `ß` (sz Ligatur)
 © = `©`
 ” = `"`;

fixes Leerzeichen = ` `; (non breakable space)

Eine komplette Übersicht gibt es unter

<http://www.w3.org/TR/html4/sgml/entities.html>

Vorsicht Falle!

Vergessen Sie nicht auf den Strichpunkt! Nur der Internet Explorer ab Version 5 zeigt Entities auch ohne Strichpunkt an.



Übung:

Ersetzen Sie in Ihren bisherigen HTML Dateien (insbes. `index.html`) alle Sonderzeichen durch die zugehörigen HTML Entities.



1.7 Textauszeichnung

Bedingt durch die Entstehungsgeschichte von HTML gibt es zwei Gruppen von Tags zur Textauszeichnung: logische Markup Tags und physikalische Markup Tags.

Logische Tags versehen ihren Inhalt mit einer Erklärung `` = hervorgehoben (*emphasised*), `` = stark hervorgehoben.

Physikalische Tags geben an wie der Inhalt dargestellt werden soll: `` = fett (**bold**), `<i>` = kursiv (*italics*).

``
``
``
`<i>`

Das ist `hervorgehobener` und das ist `besonders hervorgehobener` Text. Jetzt ist er `</i>kursiv</i>` und nun `fett`.



Abbildung 1.9: Diese Abbildung zeigt die Effekte von ``, ``, `<i>` und ``.

Die Browser stellen `` im allgemeinen kursiv und `` in fett dar (bei den ersten Browsern konnte man die Darstellung für jedes Tag einzeln wählen).

Übung:

Formatieren Sie die Namen im Künstlerlexikon (`kuenstler.html`) in fett und die Jahreszahlen in kursiv. Benutzen Sie dazu sowohl `` und `` als auch `` und `<i>`.



Ursprünglich verfügte HTML ausschließlich über Logische Markup Tags. Der Gedanke war ja, der automatisierte Informationsaustausch, der z.B. die gezielte Suche eines Namens ausschließlich in Adressfeldern ermöglichen sollte. Die optische Aufbereitung des Inhalts wurde dem Client überlassen. Die zunehmende Verbreitung von HTML führte zuerst zu einem nicht artgerechten Einsatz der bestehenden Tags und dann zu einer Fülle neuer Tags, die das Aussehen des Inhalts beschreiben wie `` oder `<i>`. Mittlerweile ist das W3C wieder davon abgekommen und diese Tags werden in den Spezifikationen nur noch als „discouraged“ (nicht empfehlenswert) oder „deprecated“ (missbilligt, abgelehnt) angeführt. Das W3C setzt nun auf die komplette Trennung von Inhalt (HTML) und Layout (CSS Cascading Style Sheets).

Ebenfalls verbunden mit dem explosionsartigem Wachstum des HTML Gebrauchs, ist ein gewisser Mangel an Konformität mit der Spezifikation. So gut wie kein Browser unterstützt den gesamten HTML4 Sprachumfang in der Weise, wie vom W3C vorgegeben. Die meisten Browser

sind ausgesprochen gutmütig, was das Interpretieren des Codes angeht. Fehlende End-Tags werden stillschweigend ergänzt, verschränkte Tags ausgebügelt, unbekanntes wird einfach ignoriert.

Der, aus der Gutmütigkeit der Browser entstehende, schlampige Programmierstil ist dem W3C offenbar ein Dorn im Auge, deshalb wurde der Nachfolger von HTML 4 XHTML1 in XML mit allen damit verbundenen Auflagen an sauberem Programmierstil definiert. Auch jetzt schon können Sie Ihren Code auf Fehler überprüfen lassen (<http://validator.w3.org/>) oder mit einem Offline Tool.

Noch ein paar Worte zum Programmierstil in diesem Skriptum: Der Stil, der hier vermittelt wird, ist sehr spezifikationsstreu. Die Attributwerte sind von Hochkommas umgeben, die End-Tags der nicht leeren Tags sind vorhanden, die Entities werden geschrieben, etc... Das ist für reines HTML vielleicht nicht nötig (schadet sicher aber auch nicht) ist aber eine Voraussetzung für den Gebrauch von Style Sheets und eine gute Vorübung für den HTML-Nachfolger XHTML (der sehr pingelig ist).