

HTML-Tutorial

1. Einleitung

1.1 Über mein Projekt

Wie ich im nächsten Kapitel zeigen werde, ist HTML heute allgegenwärtig. Sobald man eine Website zu programmieren versucht, kommt man nicht darum herum, HTML zu lernen. Sie werden mit diesem Tutorial zuerst verstehen, wieso man HTML braucht, dann werden Sie die grundlegenden Syntaxen lernen und zuletzt ihre Arbeit bei einem Gratis-Anbieter uploaden (hochladen).

Sie brauchen keine Vorkenntnisse zur Programmierung von HTML, ich werde Ihnen alles Schritt für Schritt beibringen, was Sie wissen müssen. Falls Sie sich schon in anderen Programmiersprachen auskennen - umso besser. Des Weiteren brauchen Sie auch - im Gegensatz zu vielen anderen Programmiersprachen - keine spezielle Software zur HTML-Programmierung, nur das Notepad, das auf jedem PC vorinstalliert ist, und einen Webbrowser zur Darstellung des Programms.

Am Ende der wichtigen Kapitel finden Sie in einem hellgelben Kasten ein kurzes Resümee, das Ihnen neben dem Inhalt die neu gelernten Tags und Syntaxen zusammenfasst. Dies wird Ihnen sehr behilflich sein, wenn Sie im Nachhinein einen Befehl für Ihre Webseite brauchen und ihn nicht mehr auswendig wissen.

Des Weiteren werden Sie auf Gutzeichen (▲) und Ausrufezeichen (▲) treffen. Erstere weisen auf Tipps und Tricks hin, zweite auf Gefahren.

Falls Sie irgendwelche Fragen oder Verbesserungsvorschläge zu dieser Arbeit haben, kontaktieren Sie mich per E-Mail (html-tutorial@hotmail.com) oder unter www.florianzeiter.ch.vu.

1.2 Die Geschichte des Internets

Vorwort: Das Internet ist eine Sammlung von verschiedenen Diensten, die populärsten sind das E-Mail und das WWW. Das, was wir also im Volksmund als „Im Internet surfen“ bezeichnen, würde fachlich korrekt „Im WWW surfen“ heissen.



Folgender Text wurde aufgrund der hervorragenden Qualität aus Wikipedia, der freien Enzyklopädie, herauskopiert (Artikel Internet, Kapitel Geschichte). Stand Juni 2009.

Das Internet ging aus dem im Jahr 1969 entstandenen ARPANET hervor, einem Projekt der Advanced Research Project Agency (ARPA) des US-Verteidigungsministeriums. Es wurde zur Vernetzung von Universitäten und Forschungseinrichtungen benutzt. Ziel des Projekts war zunächst, die knappen Rechenkapazitäten sinnvoll zu nutzen, erst in den USA, später weltweit. Die anfängliche Verbreitung des Internets ist eng mit der Entwicklung des Betriebssystems Unix verbunden. Nachdem das Arpanet im Jahr 1982 TCP/IP adaptierte, begann sich auch der Name Internet durchzusetzen.

Nach einer weit verbreiteten Legende bestand das ursprüngliche Ziel des Projektes vor dem Hintergrund des Kalten Krieges in der Schaffung eines verteilten Kommunikationssystems, um im Falle eines Atomkrieges eine störungsfreie Kommunikation zu ermöglichen. In Wirklichkeit wurden vorwiegend zivile Projekte gefördert, auch wenn die ersten Knoten von der ARPA finanziert wurden.

Die wichtigste Applikation in den Anfängen war die E-Mail. Bereits im Jahr 1971 überstieg das Gesamtvolumen des E-Mail-Verkehrs das Datenvolumen, das über die anderen Protokolle des Arpanet, das Telnet und FTP abgewickelt wurde.

Rasanten Auftrieb erhielt das Internet seit dem Jahr 1993 durch das World Wide Web, kurz WWW, als der erste grafikfähige Webbrowser namens Mosaic veröffentlicht und zum kostenlosen Download angeboten wurde. Das WWW wurde im Jahr 1989 im CERN (bei

Genf) von Tim Berners-Lee entwickelt. Schließlich konnten auch Laien auf das Netz zugreifen, was mit der wachsenden Zahl von Nutzern zu vielen kommerziellen Angeboten im Netz führte. Der Webbrower wird deswegen auch als die „Killerapplikation“ des Internet bezeichnet. Das Internet ist ein wesentlicher Katalysator der Digitalen Revolution.

Im Jahr 1990 beschloss die US-amerikanische National Science Foundation, das Internet für kommerzielle Zwecke zu nutzen, wodurch es über die Universitäten hinaus öffentlich zugänglich wurde.

Neue Techniken verändern das Internet und ziehen neue Benutzerkreise an: IP-Telefonie, Groupware wie Wikis, Blogs, Breitbandzugänge (zum Beispiel für Vlogs und Video-on-Demand), Peer-to-Peer-Vernetzung (vor allem für File Sharing) und Online-Spiele (z. B. Rollenspiele, Taktikshooter, ...).



Eine chronologische Zusammenfassung der Geschichte des Internets
finden Sie auf Wikipedia unter
http://de.wikipedia.org/wiki/Chronologie_des_Internets

1.3 HTML im Alltag und Webbrowser

HTML ist englisch und steht für Hypertext Markup Language (dt. Hypertext-Auszeichnungssprache). HTML-Dokumente sind die Grundlage des World Wide Web. HTML ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten (Zitat aus Wikipedia). So genannte Webbrowser (die populärsten sind Mozilla Firefox, Windows Internet Explorer, Opera und Safari) sind in der Lage, HTML-Skripte (in HTML geschriebene Dokumente) anzuzeigen.

Da sich Firefox auf einem aufstrebenden Ast befindet und den Internet Explorer Anfangs 2009 als führenden Webbrower abgelöst hat (was zuvor noch kein Browser geschafft hat), habe ich mein Tutorial grösstenteils mit Mozilla Firefox entwickelt und getestet. Der grösste Unterschied zwischen Internet Explorer und Mozilla Firefox besteht in den Add-Ons, die nur

von Firefox angeboten werden. Add-Ons sind kleine, direkt in den Browser integrierte Zusatzprogramme, die das Surfen angenehmer, interessanter und einfacher machen. Mozilla bietet unter <https://addons.mozilla.org/de/firefox> mehr als hundert Millionen „offizielle“ und garantiert virenfreie Add-Ons an, die direkt heruntergeladen werden können und automatisch in den Browser integriert werden.

Statistiken zufolge wird Mozilla grösstenteils von jüngeren, Internet Explorer von älteren Personen gebraucht. Dies liegt wahrscheinlich an der Einstellung der User zum Computer. Während ältere Personen häufig zufrieden sind, wenn alles so läuft, wie es sein sollte und sich nicht für Neues aus dem Informatik-Bereich interessieren, wollen Jugendliche auch Alternativen in Betracht ziehen, um den idealen Webbrower herauszufinden. Dass so viele Anwender nicht falsch liegen können, beweist auch folgende Statistik, die von diversen bekannten Tester (u.a. www.chip.de) bestätigt wurde: „Der Firefox überflügelt den Internet Explorer in den meisten Disziplinen: Er ist sicherer, besser ausgestattet und lässt sich flexibel erweitern.“ Obwohl ausser Internet Explorer praktisch alle bekannten Webbroser Macintosh-kompatibel sind, empfiehlt sich für Mac eigentlich nur Safari, der von Werk aus auf Ihrem Betriebssystem installiert ist. Der grösste Teil meiner Arbeit gilt sowohl für Windows als auch für Macintosh, Abweichungen sind markiert.

Falls Sie auf einen andern Browser wechseln wollen oder noch gar keinen installiert haben, finden Sie die Downloadlinks auf den jeweiligen Herstellerseiten:

<http://www.mozilla-europe.org/de/> (Mozilla Firefox)

<http://www.microsoft.com/germany/windows/internet-explorer/download-ie.aspx> (IE)

<http://de.opera.com/> (Opera)

<http://www.apple.com/de/safari/> (Apple Safari)

Wie Sie wahrscheinlich schon bemerkt haben, empfehle ich Ihnen aufgrund der Sicherheit und den individuellen Erweiterungsmöglichkeiten Mozilla Firefox.

Wenn wir also mit einem Webbrower auf Google gehen, dann liest der Browser den Quelltext (das in HTML geschriebene Dokument) und gibt ihn grafisch aus. Anders als zum Beispiel in Word können wir in HTML nicht einfach ein Bild ins Programm hineinziehen, wir müssen einen Befehl programmieren, der dem Browser sagt, dass er jetzt dieses Bild hier

einfügen soll. Es bedarf also einiger Versuche, bis wir es schaffen, den Quelltext so zu programmieren, dass das Resultat im Browser unseren Vorstellungen entspricht.

Damit Sie sich das vorstellen können, folgt ein Beispiel, wie ein Webbrower (hier Internet Explorer) ein HTML-Dokument (im Editor geschrieben) ausgibt:



```
Beispiel - Editor
Datei Bearbeiten Format Ansicht ?
<html>
  <head>
    <title>Beispiel</title>
  </head>
  <body bgcolor = "yellow">
    <h1>Beispiel für HTML</h1>
    <p>
      Dies ist ein Beispiel zur Demonstration,
      wie ein Webbrower <br>
      ein HTML-Dokument in eine Website
      umwandelt.
    </p>
    <a href="http://www.google.ch">Google</a>
  </body>
</html>
```

Abbildung 1.3.1: Website im Notepad

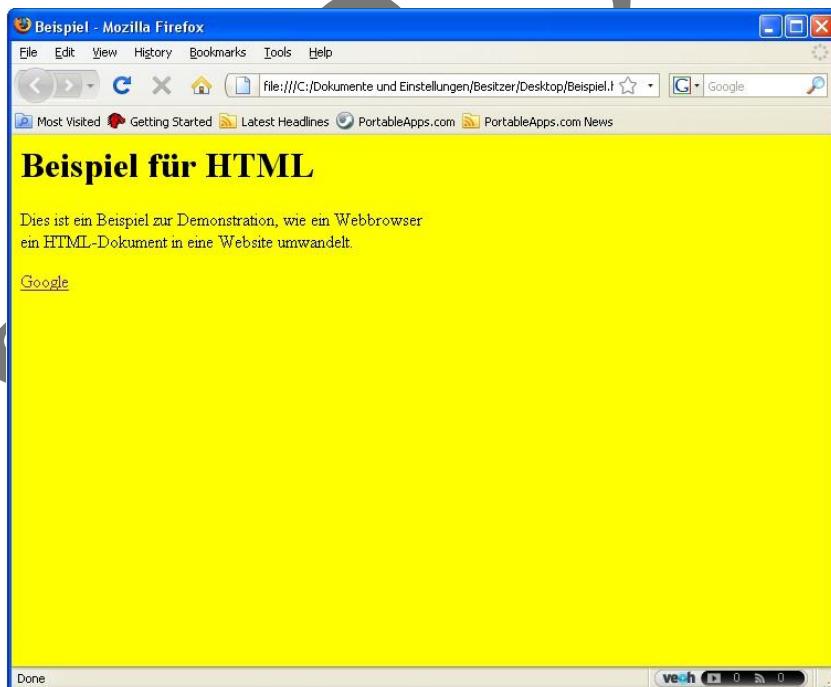


Abbildung 1.3.2: Gleiche Website im Webbrower (hier Mozilla Firefox)

Wie jede Programmiersprache wurde auch HTML aus einer Sprache entwickelt und wurde schon zu einer neueren weiterentwickelt. HTML wurde der Nachfolger von SGML und ist die Grundlage für XHTML, die neuere Version von HTML. Zeitweilig wurde mit dem Gedanken gespielt, dass XHTML HTML ablösen soll, was aber nie eintrat. XHTML gleicht HTML sehr stark, es gibt aber einige Unterschiede, beispielsweise wird in XHTML im Gegensatz zu HTML Gross- und Kleinschreibung unterschieden und das Weglassen von End-Tags (mehr dazu später) ist nicht erlaubt.

Sie haben sicherlich schon Werbung von zum Beispiel Bluewin oder vielen anderen grossen Internet-Firmen gesehen, wo Sie gratis in wenigen Schritten eine eigene Website erstellen können. Das ist sicherlich viel einfacher, als eine eigene Website zu programmieren, aber man sollte sich dann überlegen, wieso man überhaupt eine Website will. Wenn es nicht der Spass am Programmieren ist, so wahrscheinlich die Internetpräsenz. Man kann aber mit diesen Internetbaukästen nur gewisse Vorlagen verwenden. Wenn man selber kreativ sein will und die eigenen Ideen umzusetzen versucht, merkt man schnell, dass man mit diesen Vorlagen schnell am Limit ist. Wenn Sie dieses Tutorial abgeschlossen haben, bin ich davon überzeugt, dass Sie auch in der Lage sind, eine Website nach Ihren Wünschen zu gestalten, ohne sich über die Beschränkungen der Online-Vorlagen nerven zu müssen.

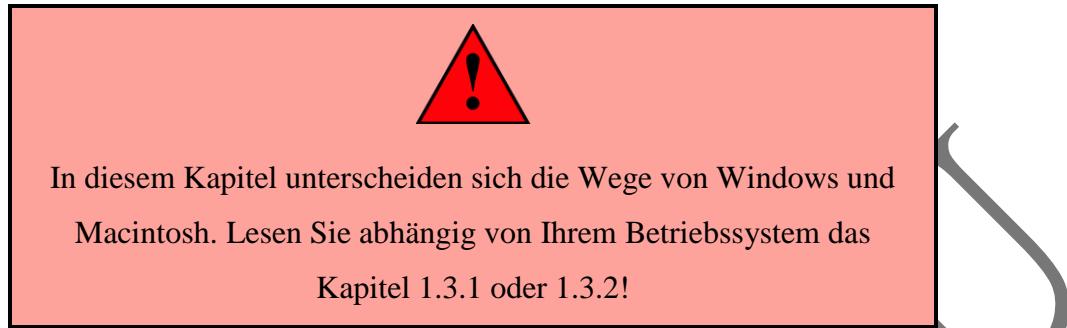


Um auf einer beliebigen Website mit der Endung .html oder .htm den Quelltext betrachten zu können (vielleicht, um noch etwas zu lernen), klicken Sie mit der rechten Maustaste irgendwo auf das Anzeigefenster des Browsers und dann auf Quelltext anzeigen.

1.4 Grundlegendes für HTML

Wie schon erwähnt, werden HTML-Dokumente in sogenannten Text-Editoren programmiert. Bedingt durch die Bequemlichkeit der Benutzer wurden sogenannte WYSIWYG-Editoren (What You See Is What You Get) entwickelt. Wie es der Name schon sagt, wird ein in einen WYSIWYG-Editor hineingeschriebenen Text mitsamt der Formatierung automatisch in reines html umgeschrieben. Bei den neueren WYSIWYG-Editoren ist es sogar möglich, parallel den

Quelltext in html und das Aussehen im Browser zu editieren. Gängige kostenpflichtige WYSIWYG-Editoren sind DreamWeaver und GoLive. Dieses Tutorial wird sich ausschliesslich mit dem Notepad (also keinem WYSIWYG-Editor) beschäftigen.



1.4.1 Betriebssystem Windows (95, 98, ME, 2000, XP, Vista oder 7)

Um dem Browser mitteilen zu können, was er anzeigen soll, müssen Sie ihm das in einem sogenannten Texteditor-Programm hinschreiben. Ein dazu geeignetes und auf jedem PC installiertes Programm ist das Notepad (zu deutsch Notizblock). Um das Notepad zu starten, gehen Sie auf Start, Programme, Zubehör, Editor oder geben Sie im Ausführen-Feld (Start, Ausführen) notepad ein. Wenn Sie ihr Programm geschrieben haben, klicken Sie auf Datei, Speichern unter. Würden Sie die Datei ganz normal abspeichern, hätten Sie ein ganz normales Textdokument und kein HTML-Dokument. Geben Sie also anstelle nur des Dateinamens den Namen und zusätzlich die Endung .html ein (also Beispiel.html). Fakultativ können Sie auch den Dateityp zu Alle Dateien ändern, um einen besseren Überblick zu bekommen.

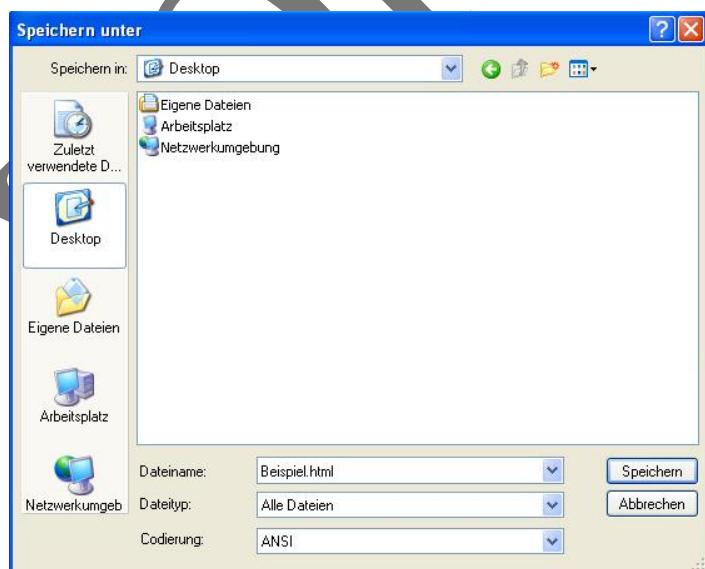


Abbildung 1.4.1: Abspeichern einer Website im Notepad, dem weitverbreitesten Texteditor

Nachdem Sie das Programm abgespeichert haben, erscheint bei dem von Ihnen gewählten Speicherort eine Datei mit dem Symbol Ihres Webbrowsers. Doppelklicken Sie nun auf diese Datei, um sie mit dem Browser anzuzeigen. Sind Sie mit dem Quelltext noch nicht zufrieden, klicken Sie mit der Rechten Maustaste auf die Datei, dann auf Öffnen mit, dann Editor. Ändern Sie die betreffenden Stellen und speichern Sie die Datei anschliessend wieder ab.

Erstellen eines HTML-Dokuments unter Windows:

1. Öffnen des Notepads (Start, Ausführen, notepad)
2. Schreiben des HTML-Skripts
3. Datei, Speichern unter
4. NameDesProgramms.**html** im Feld Dateiname
5. Speichern
6. Doppelklick auf Datei zum Öffnen mit dem Browser
7. Rechtsklick auf Datei, Öffnen mit, Editor zum Öffnen und Korrigieren mit dem Editor



Wenn Sie ein anspruchsvolles und langes Programm schreiben wollen und nicht immer den Editor und den Browser neu starten wollen, lassen Sie den Editor und den Browser parallel laufen. Wenn Sie im Editor Änderungen vorgenommen haben, speichern sie das Dokument ab (ctrl + s) und wechseln Sie zum Browser. Aktualisieren Sie ihn durch drücken von F5. Das Aktualisieren ist nötig, da der Browser die geladene Seite im Cache "Puffer-Speicher" behält und nicht aktiv die neuste Version lesen geht.

1.4.2 Betriebssystem Macintosh (OS 1-9 oder OS X)

Um dem Browser mitteilen zu können, was er anzeigen soll, müssen Sie ihm das in einem sogenannten Texteditor-Programm hinschreiben. Um den Texteditor zu starten, klicken Sie auf Finder, Programme, Texteditor. Klicken Sie in der Befehlszeile auf Format, „In reinen

Text umwandeln“, damit der Editor die Datei als Website abspeichern kann. Um die Datei abzuspeichern, klicken Sie auf Ablage, Sichern unter. Geben Sie den Dateinamen in der Form gewünschterDateiname.html ein, wobei Sie „gewünschterDateiname“ natürlich verändern können. Die Endung .html lässt Ihr Betriebssystem wissen, dass es sich um eine Safari-Anwendung handelt (wenn Sie Safari als Ihr Standartbrowser eingestellt haben) und sie mit Safari geöffnet werden muss.

Nachdem Sie das Programm abgespeichert haben, erscheint bei dem von Ihnen gewählten Speicherort eine Datei mit dem Symbol Ihres Webbrowsers. Doppelklicken Sie nun auf diese Datei, um sie mit dem Browser anzuzeigen. Sind Sie mit dem Quelltext noch nicht zufrieden, öffnen Sie den Texteditor (siehe oben), klicken Sie auf Datei, Öffnen und wählen Sie Ihre Datei aus. Ändern Sie die betreffenden Stellen und speichern Sie die Datei anschliessend wieder ab.

Erstellen eines HTML-Dokuments unter Macintosh:

1. Öffnen des Texteditors (Finder, Programme, Texteditor)
2. Format, In reinen Text umwandeln
3. Schreiben des HTML-Skripts
4. Ablage, Sichern unter
5. gewünschterDateiname.**html** im Feld Dateiname
6. Speichern
7. Doppelklick auf Datei zum Öffnen mit dem Browser

2. Programmierung

2.1 Allgemeine Regeln für HTML

Im Kapitel Eins haben Sie gelernt, wieso man HTML zur Programmierung von Webseiten braucht und wieso es wenig sinnvoll ist, Online-Baukästen zur Erstellung einer Webseite zu brauchen. Des Weiteren wissen Sie, mit welchem Programm man ein HTML-Skript schreibt und wie man es richtig abspeichert. Nun wollen wir uns der HTML-Programmierung zuwenden.

Wie Sie vielleicht schon aus der Grafik im Kapitel 1.2 entnommen haben, enthält jedes HTML-Dokument so genannte Tags (die Wörter mit den spitzen Klammern, z.B. `<html>`). Tags (engl. sprich tag) sind Befehle, zum Beispiel gibt es den HTML-Tag am Anfang des Dokuments, den HEAD- und BODY-Tag und so weiter. Jeder dieser Tags muss mit dem Start-Tag und einem Schrägstrich (/) nach dem Kleiner-Als-Zeichen (z.B. `</html>`) beendet werden. Der HTML-Tag wird zum Beispiel erst am Ende des Programms mit `</html>` beendet. Der HEAD-Tag hingegen wird schon viel früher mit `</head>` beendet. Im Allgemeinen beschränkt der Anfangs- mit dem End-Tag den Gültigkeitsbereich des Tags. Der HTML-Tag beginnt zu Beginn des Programms, das Ende ist ganz am Schluss des Programms, während ein Textformatierungstag wie zum Beispiel `<i>` (kursiv) mit `</i>` den Bereich definiert, der kursiv gedruckt sein soll. Natürlich ist es auch möglich, Tags ineinander zu verschachteln, wenn Sie zum Beispiel im BODY-Tag einen Text mit `<i>` kursiv gedruckt haben wollen. Sie werden im Verlaufe dieses Tutorials viele Tags lernen müssen, um eine Website nach Ihren Wünschen gestalten zu können.

Zu jedem Tag gibt es diverse sogenannte Attribute, die die Funktion des Tags genauer eingrenzen. Wenn wir zum Beispiel einen Link definieren wollen, lautet die Syntax `<a>Link`. Mit dem Attribut `href="http://www.google.ch"` wird programmiert, dass der Link den User auf Google führen soll. Mit dem Attribut `target = "_blank"` wird der Link in einem neuen Fenster geöffnet. Oftmals sind für einen Tag noch mehr Attribute verfügbar; in diesem Tutorial sind nur die Wichtigsten aufgelistet, damit man den Überblick nicht verliert. Weitere Attribute können Sie im WWW nachschlagen, zum Beispiel unter <http://de.selfhtml.org/html/index.htm>.

Hier Klicken

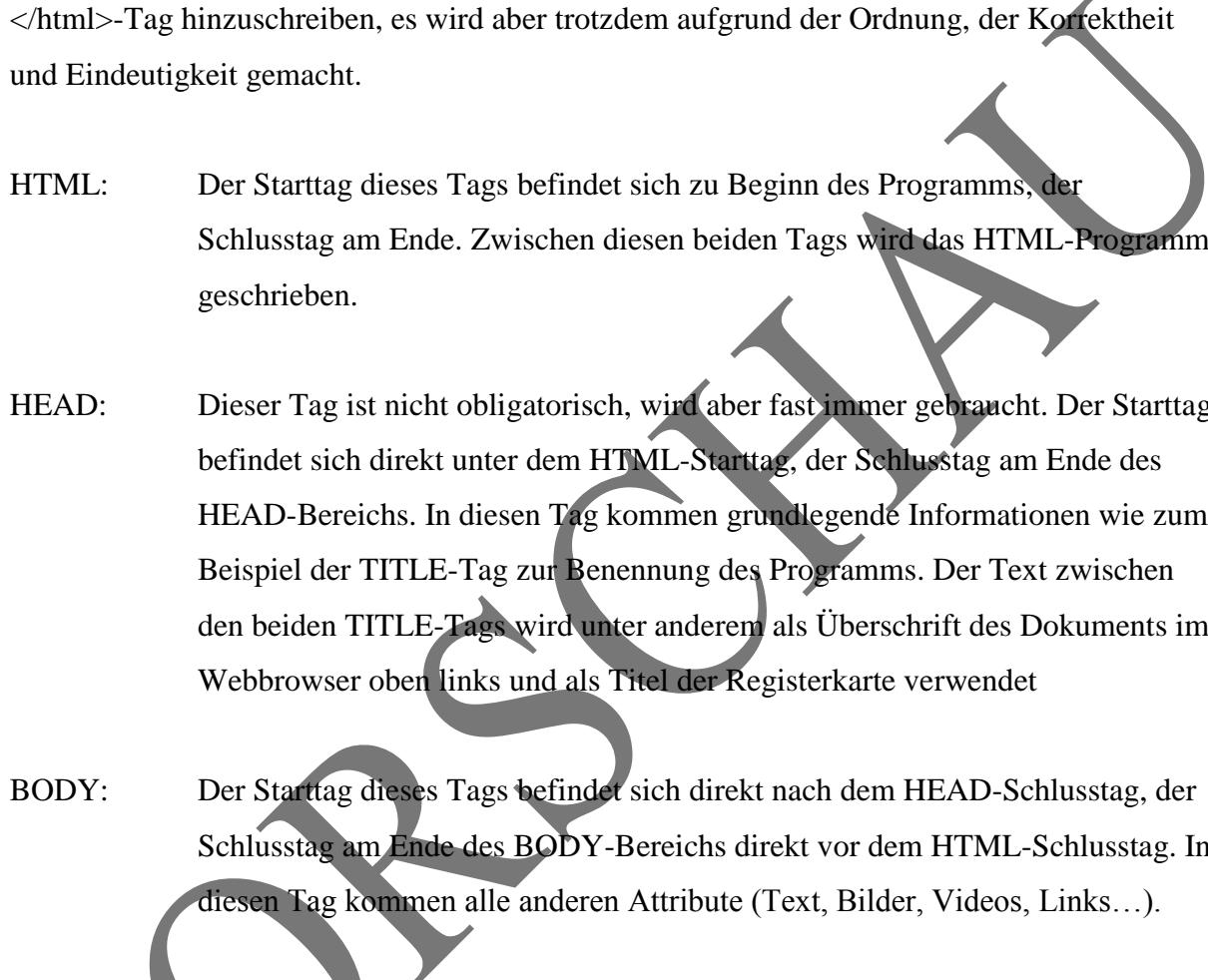
Abbildung 2.1.1: Tag (Anfangs- und Schlusstag in Rot) mit Attributen (Grüntöne) und dem angezeigten Text (Blau)

Die drei grundlegendsten Tags sind der HTML-, der HEAD- und der BODY-Tag. Zum Beispiel beim HTML-Tag ist es nicht nötig, am Ende des Programms den entsprechenden </html>-Tag hinzuschreiben, es wird aber trotzdem aufgrund der Ordnung, der Korrektheit und Eindeutigkeit gemacht.

HTML: Der Starttag dieses Tags befindet sich zu Beginn des Programms, der Schlusstag am Ende. Zwischen diesen beiden Tags wird das HTML-Programm geschrieben.

HEAD: Dieser Tag ist nicht obligatorisch, wird aber fast immer gebraucht. Der Starttag befindet sich direkt unter dem HTML-Starttag, der Schlusstag am Ende des HEAD-Bereichs. In diesen Tag kommen grundlegende Informationen wie zum Beispiel der TITLE-Tag zur Benennung des Programms. Der Text zwischen den beiden TITLE-Tags wird unter anderem als Überschrift des Dokuments im Webbrowser oben links und als Titel der Registerkarte verwendet

BODY: Der Starttag dieses Tags befindet sich direkt nach dem HEAD-Schlusstag, der Schlusstag am Ende des BODY-Bereichs direkt vor dem HTML-Schlusstag. In diesen Tag kommen alle anderen Attribute (Text, Bilder, Videos, Links...).



```
Unbenannt - Editor
Datei Bearbeiten Format Ansicht ?
<html>
    <head>
        <title>Der Titel des Dokuments</title>
    </head>
    <body>
        Dies ist ein Text.
    </body>
</html>
```

Abbildung 2.1.2: Grundlegendes HTML-Gerüst (auch HTML-Syntax genannt) im Notepad

Jedes HTML-Dokument besteht aus mindestens drei Tags, dem HTML, dem HEAD und dem BODY-Tag. Während der erste dazu dient, den Start und das Ende des Programmes zu markieren, werden im zweiten Metadaten wie zum Beispiel der Titel der Website definiert und im dritten wird der eigentliche Inhalt hingeschrieben.



Im Quelltext spielt das Layout theoretisch keine Rolle. Wenn Sie zum Beispiel den Quelltext der Google-Startseite betrachten (Rechtsklick, Quelltext anzeigen), werden Sie nur einige sehr lange Zeilen vorfinden. Wenn Sie aber als Normalanwender eine Webseite programmieren wollen und vielleicht später noch einige Änderungen vornehmen werden, empfiehlt es sich, von Abständen (mit Tabulator), neuen Zeilen und Kommentaren (siehe unten) Gebrauch zu machen, um sich im Quelltext wieder zurechtfinden zu können.



Wenn Sie ein langes Programm schreiben oder auch Javascripts einbetten wollen, empfiehlt es sich, Kommentare zu brauchen. Kommentare sind Informationen des Programmierers, die vom Browser nicht interpretiert und in der Webseite nicht angezeigt werden. Wenn man also zum Beispiel eine Navigationsleiste programmiert, sollte man zur besseren Übersicht vor dem Start der Leiste einen Kommentar einfügen, dass hier die Leiste beginnt und nachher, dass sie hier aufhört. Kommentare schreibt man mit <!-- DiesIstEinKommentarDerNichtInDerWebsiteAngezeigtWird-->. Das Ausrufezeichen nach dem Kleiner-Als-Zeichen und die zwei Bindestriche zu Beginn und am Ende des Kommentars dürfen natürlich nicht vergessen werden.

2.2 Meta-Angaben

Meta-Angaben werden in den HEAD-Tag des Programms eingefügt. Es sind Informationen für den Nutzer, zum Beispiel, wer das Programm geschrieben hat, oder für die Internetsuchmaschinen, um die Webseite schnell finden zu können. Die Syntax dazu ist <meta name = "Attribut" content = "Wert">, also zum Beispiel <meta name = "author" content = "Franz Muster">. Hier eine Liste der am Weitesten verbreiteten Meta-Angaben:

<meta name = "author" content = "Franz Muster"> Hiermit wird der Name des Autors angegeben.

<meta name = "description" content = "Meine Website enthält Bilder vom Lothar"> Diese Beschreibung erscheint einem Anwender im Suchdienst beim Auffinden dieser Datei.

<meta name = "keywords" content = "Bilder, Lothar, Orkan, 1999"> Diese Begriffe sind Stichwörter für ein Suchprogramm, zum Beispiel Google.

<meta name = "date" content = "2001-12-15T08:49:37+02:00"> Die Zeiteingabe vom Programmierer (wann er diese Webseite publiziert hat) sollte in diesem standartisierten Schema erfolgen. Im Beispiel ist dies der 15. Dezember 2001 um 8.49.37 Uhr. Die Angabe hinter dem Pluszeichen ist die Abweichung von UTC (hier Mitteleuropäische Sommerzeit).

<meta name = "robots" content = "noindex"> Mit diesem Befehl verbieten Sie den Robots von Suchmaschinen, diese HTML-Datei in ihre Datenbank einzufügen, damit ein Benutzer sie bei Abfrage in den Suchresultaten nicht sehen kann. Robots sind Programme von Suchmaschinen, die das WWW ständig nach neuen oder aktualisierten Webseiten durchforsten. Sie können alternativ auch alle Meta-Angaben löschen, dann kann keine Suchmaschine Ihre Webseite finden.

<meta http-equiv = "refresh" content = "5; URL=http://www.google.ch"> Fünf Sekunden nach dem Öffnen der Datei wird der User automatisch auf Google weitergeleitet. Anstelle einer Internetadresse können Sie auch den Pfad einer eigenen Datei angeben (mehr dazu unter Links). Dieser Befehl hat die gleiche Wirkung, wie wenn der Benutzer nach fünf Sekunden die URL http://www.google.ch in die Browserzeile eingibt und mit Enter bestätigt.

Die im HEAD-Tag geschriebenen Meta-Angaben dienen Nutzern und Internetsuchmaschinen, Informationen wie zum Beispiel der Autor der Website oder Keywords zu erfassen.

```
<meta name = "author" content = "Franz Muster">  
<meta name = "description" content = "Bilder vom Lothar">  
<meta name = "keywords" content = "Bilder, Lothar, Orkan, 1999">  
<meta name = "date" content = "2001-12-15T08:49:37+02:00">  
<meta name = "robots" content = "noindex">  
<meta http-equiv = "refresh" content = "5; URL=www.google.ch">
```

2.3 Textformatierung

Wie Sie schon in 2.1 gelernt haben, spielt es keine Rolle, wann und ob Sie überhaupt Abstände und neue Zeilen im Notepad brauchen. Dass dies aber trotzdem vorteilhaft ist, ist ja logisch. Wenn wir hingegen einen Text veröffentlichen, wollen wir sicher ab und zu eine neue Zeile beginnen oder einen kleinen Absatz programmieren. Dazu gibt es natürlich Tags (was ein Tag ist, wissen Sie ja schon). Die ganze Textstrukturierung befindet sich im BODY-Tag. Hier eine Liste der wichtigsten Tags zur Textformatierung.

2.3.1 Neue Zeile

Mit dem `
`-Tag (engl. break) können Sie eine neue Zeile beginnen, was im Word etwa dem einmaligen Drücken der Eingabe-Taste entspricht. Obwohl Tags zwar immer einen Start- und Schlusstag haben sollten, kann beim `
`-Tag der Schlusstag weggelassen werden.

2.3.2 Absatz

Mit dem `<p>`-Tag (engl. paragraph) können Sie einen Absatz machen, was im Word etwa dem zweimaligen Drücken der Eingabe-Taste entspricht. Auch dieser Tag muss nicht unbedingt geschlossen werden.

2.3.3 Mehrere Leerzeilen

Manchmal reicht ein einzelner Absatz nicht aus; man will zwischen zwei Textblöcken (oder zwischen Bildern) einen grösseren Abstand. Es bringt dann nichts, mehrere `<p>`-Tags aneinanderzuketten. Dazu gibt es aber eine Lösung in Form des `<pre>`-Tags. Drücken Sie zwischen dem Start- und dem Schlusstag (der ist hier natürlich nötig) beliebig viele Male die Eingabetaste. Der Abstand im Quelltext von `<pre>` zu `</pre>` entspricht auch dem Abstand zwischen den beiden sich vor und nach den Tags befindenden Objekten im Browser. Dies ist also einer der wenigen Befehlen, wo sich das Layout im Quelltext auf die Anzeige im Browser auswirkt!

2.3.4 Geschütztes Leerzeichen

Ab und zu ist es von Vorteil, zwischen zwei Wörtern mehrere Leerzeichen zu haben. Wenn Sie aber im Quelltext mehrere Leerzeichen aneinanderketten, bringt das genauso wenig, wie wenn sie für eine neue Zeile einmal auf die Eingabe-Taste drücken. Sie müssen dazu ein so genanntes Geschütztes Leerzeichen verwenden. Der Befehl dazu ist ` ` und sie können jetzt beliebig viele Leerzeichen aneinanderreihen. Wollen Sie zum Beispiel zwischen den Wörtern Beispiel und Hallo drei Leerzeichen, schreiben Sie Beispiel ` ` ` ` ` ` Hallo.

2.3.5 Überschriften

Sicherlich wollen Sie auch Überschriften schreiben, die grösser sein sollen als der restliche Text. Dazu können sie den `<hx>`-Tag benutzen. Das x steht für eine beliebige Zahl zwischen 1 und 6, wobei eins die grösste Schrift ist und sechs die kleinste. Zusammen mit dem Schlusstag ist der Bereich zwischen den beiden Tags in dieser Schriftgrösse. Zum Beispiel ist `<h2>DieserText</h2>` in der zweitgrössten Schriftgrösse, was etwa der Grösse der Überschrift 2.3 entspricht.

2.3.6 Kursiv und Fett

Manchmal sollen Texte auch kursiv oder fett sein. Der Tag für kursiv ist `<i>` (engl. italic) und muss mit `</i>` beendet werden, wobei der Text zwischen den beiden Tags kursiv gedruckt ist.

Parallel zu *<i>* gibt es **** (engl. bold), der auch durch ** abgeschlossen werden muss. Sie können natürlich auch mit *<i>DiesenTextHier</i>* fett und kursiv drucken. Zum Beispiel wird *<i>Hallo</i>* im Browser als *Hallo* angezeigt.

2.3.7 Unter- und durchstreichen, hoch- und tiefstellen

Mit <u> und *</u>* wird der Text unterstrichen, mit ~~<strike>~~ und *</strike>* durchgestrichen, mit <big> und *</big>* grösser, mit <small> und *</small>* kleiner, mit ^{und *}* hoch- und mit _{und *}* tiefgestellt. Zum Beispiel wird <u>Hallo</u> im Browser als Hallo angezeigt.

2.3.8 Textausrichtung

Sie können Texte - wie im Word - auch rechtsbündig und zentriert ausrichten. Die Befehle dazu sind *<div align = "right">* und *</div>* für rechtsbündig und *<div align = "center">* und *</div>* für zentriert ausgerichtet.

2.3.9 Textfarbe

Vielleicht wollen Sie eine Textstelle besonders hervorheben, zum Beispiel, indem Sie ihn rot schreiben. Dazu brauchen Sie den FONT-Tag. Mit *IstDieserText* rot geschrieben. Anstelle von red können Sie auch die RGB-Werte angeben (siehe Hintergrundfarbe). Zum Beispiel wird *Hallo* im Browser als *Hallo* angezeigt.

2.3.10 Hintergrundfarbe

Finden Sie einen weissen Hintergrund langweilig? Sie können den Hintergrund auf zwei verschiedene Arten färben. Mit dem Befehl *<body bgcolor = "yellow">* wird der Hintergrund Gelb. Es gibt sehr viele solche vorprogrammierte Farben. Eine Liste der wichtigsten finden Sie auf www.html-tutorial.ch.vu. Man kann die Hintergrundfarbe aber auch mit dem Befehl *<body bgcolor = "#0000FF">* definieren, welche in diesem Fall Blau wäre. Wie man zu dieser hexadezimalen Angabe von Farben gekommen ist, können Sie unter www.html-tutorial.ch.vu nachlesen. Eine Liste vieler solcher RGB-Codes mit der dazugehörenden Farbe finden Sie

unter www.html-tutorial.ch.vu. Falls Sie ein Bild aus einer Datei als Hintergrund wollen, brauchen Sie den Befehl `<body background = "Hintergrund.jpg">`, wobei das Bild `Hintergrund.jpg` im gleichen Ordner wie das HTML-Dokument sein muss.

2.3.11 Schriftart

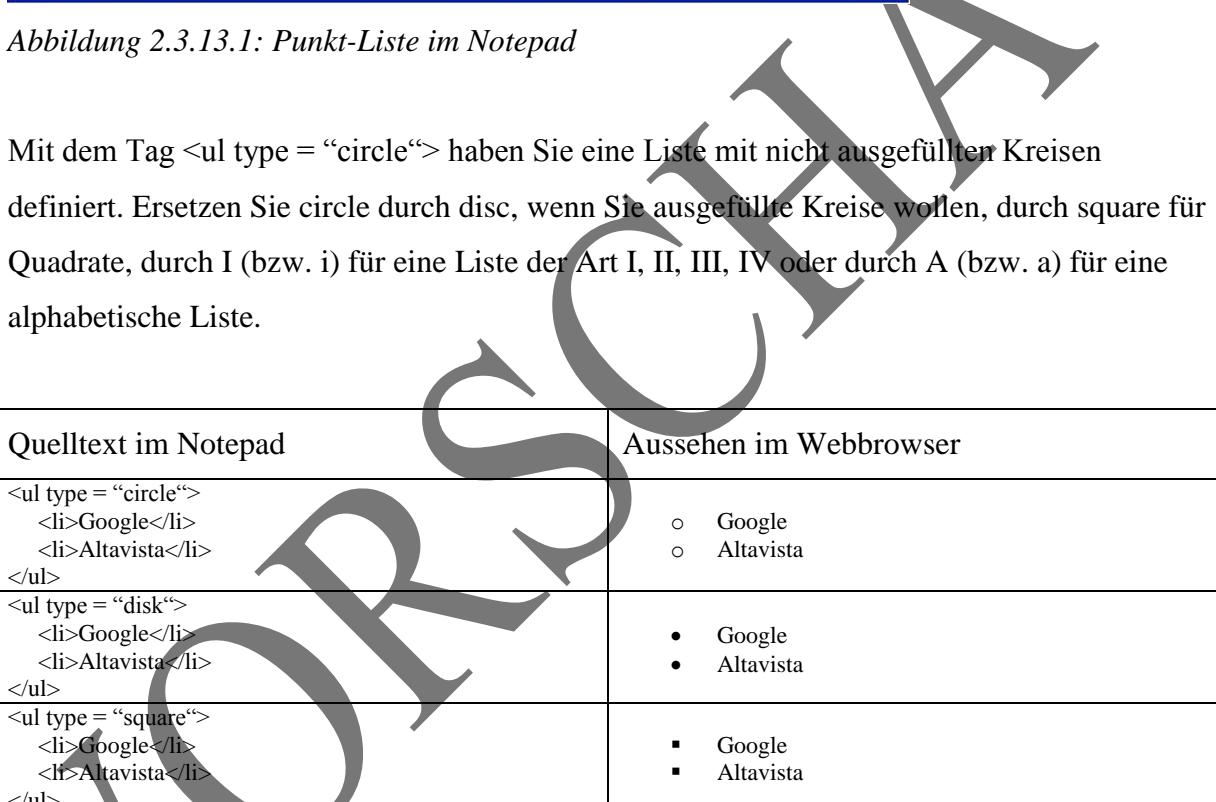
Vielleicht wollen Sie eine bestimmte Textstelle in einer anderen Schriftart schreiben? Auch dazu brauchen Sie den FONT-Tag: `Schriftart Avalon`. Zum Beispiel wird `Hallo` im Browser als `Hallo` angezeigt.

2.3.12 Lauftext

Sie können auch einen Lauftext programmieren. Ein Lauftext ist ein Text, der von links nach rechts (oder umgekehrt) vorbeizieht. Mit dem Befehl `<marquee scrollamount = "20" direction = "right">` und `</marquee>` können Sie an einem beliebigen Ort im Quelltext (meistens passt ein Lauftext nur zu Beginn oder am Ende des Programms) einen Lauftext einfügen, wobei scrollamount die Geschwindigkeit in Pixeln pro Millisekunde und direction die Richtung (right = Laufrichtung nach rechts; left = Laufrichtung nach links) ist. Als drittes Attribut können Sie vor dem Grösser-Als-Zeichen des Starttags den Befehl `behavior = "alternate"` einfügen, damit der Text von links nach rechts und dann wieder nach links geht.

2.3.13 Listen

Sie können in HTML zwischen verschiedenen Listen auswählen. Natürlich können Sie auch Listen ineinander verschachteln. Der Tag zum Start einer normalen Punkt-Liste ist `<ul type = "circle">` und der Schlusstag ``. Sie können nun mit dem Befehl `` und `` beliebig viele Punkte zwischen die beiden ``-Tags einfügen:



```

<html>
  <body>
    <ul type = "circle">
      <li>Google</li>
      <li>Altavista</li>
      <li>Fireball</li>
    </ul>
  </body>
</html>

```

Abbildung 2.3.13.1: Punkt-Liste im Notepad

Mit dem Tag `<ul type = "circle">` haben Sie eine Liste mit nicht ausgefüllten Kreisen definiert. Ersetzen Sie circle durch disc, wenn Sie ausgefüllte Kreise wollen, durch square für Quadrate, durch I (bzw. i) für eine Liste der Art I, II, III, IV oder durch A (bzw. a) für eine alphabetische Liste.

Quelltext im Notepad	Aussehen im Webbrowser
<pre><ul type = "circle"> Google Altavista </pre>	<ul style="list-style-type: none"> ○ Google ○ Altavista
<pre><ul type = "disk"> Google Altavista </pre>	<ul style="list-style-type: none"> ● Google ● Altavista
<pre><ul type = "square"> Google Altavista </pre>	<ul style="list-style-type: none"> ▪ Google ▪ Altavista
<pre><ul type = "I"> Google Altavista </pre>	<ol style="list-style-type: none"> I. Google II. Altavista
<pre><ul type = "A"> Google Altavista </pre>	<ol style="list-style-type: none"> A. Google B. Altavista

Eine nummerierte Liste wird durch `` und `` beschränkt (kein type!). Einzelne Aufzählungspunkte werden mit `` und `` gekennzeichnet:

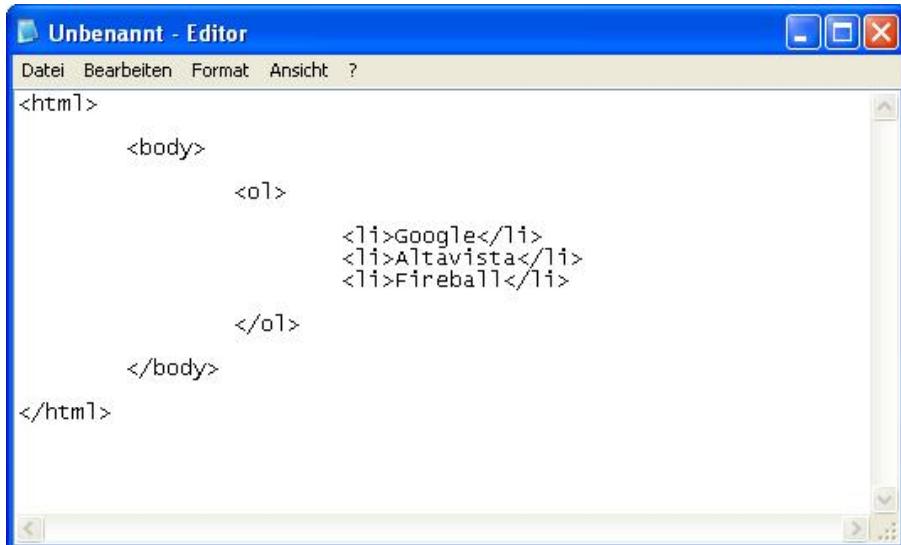


Abbildung 2.3.13.2: Nummerierte Liste im Notepad

Quelltext im Notepad	Aussehen im Webbrower
<pre> Google Altavista </pre>	<p>1. Google 2. Altavista</p>

2.3.14 Präformatierter Text

Wie Sie schon früher gelernt haben, erscheint zwischen den beiden `<pre>` bzw. `</pre>`-Tags der Inhalt des Quelltext genau so im Browser (=präformatierter Text). Dass dies aber nicht nur dazu gebraucht werden kann, mehrere Leerzeilen zwischen zwei Textblöcken einzufügen, versteht sich von selbst:

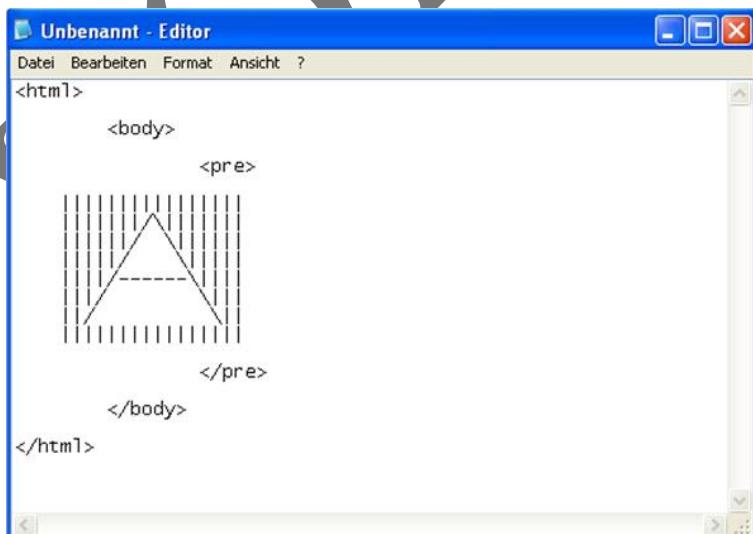


Abbildung 2.3.14.1: Grosses A als präformatierter Text im Notepad

2.3.15 Trennlinien

Ab und zu ist es sinnvoll, zur optischen Auflockerung eine Trennlinie einzufügen. Der Befehl dazu ist `<hr>` (engl. horizontal rule), der Schlusstag ist nicht obligatorisch. Sie können Trennlinien auch formatieren - die Grösse, die Dicke und die Ausrichtung ändern. Zum Beispiel ist `<hr width = "300" size = "3" align = "center">` eine dreihundert Pixel lange und drei Pixel dicke zentrierte Trennlinie.

Mit Textformatier-Tags können Textformatierungen an der Website vorgenommen werden:

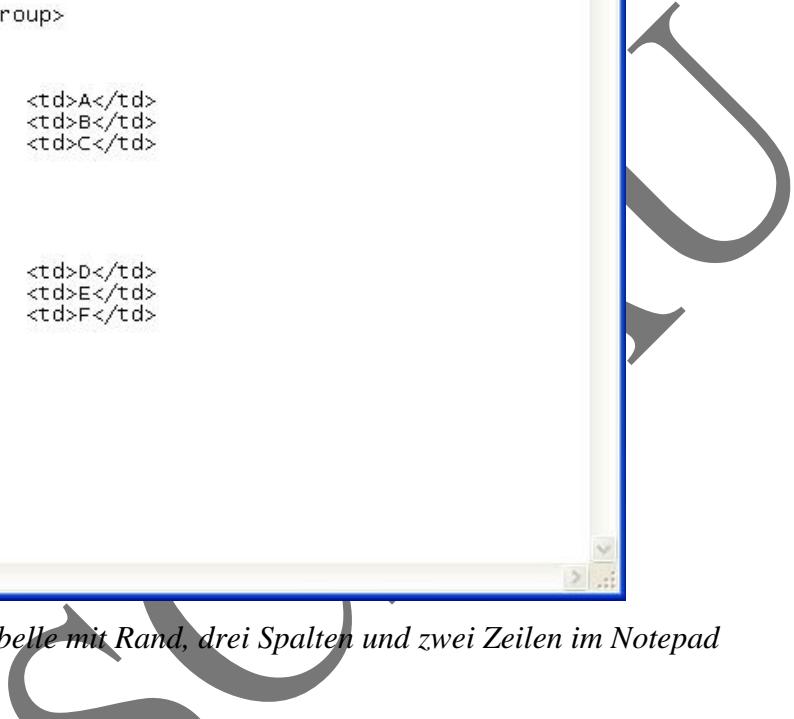
```
<br> Neue Zeile  
<p> Paragraph bzw. grosser Abstand  
<pre> Präformatierter Text  
&#160; Geschütztes Leerzeichen  
<h1> bis <h6> Überschriften  
<i> Kursiv  
<b> Fett  
<u> Unterstrichen  
<strike> Durchgestrichen  
<sup> Hochgestellt  
<sub> Tiefgestellt  
<div align = "right"> Textausrichtung (hier rechts)  
<font color = "red"> Textfarbe (hier rot)  
<body bgcolor = "yellow"> Hintergrundfarbe (hier gelb)  
<body bgcolor = "#0000FF> Hintergrundfarbe (hier blau)  
<body background = "Hintergrund.jpg"> Hintergrundbild  
<font face = "Avalon"> Schriftart (hier Avalon)  
<marquee scrollamount = "20" direction = "right"> Lauftext  
<ul type = "circle"> und <li> Punktliste  
<ol> und <li> Nummerierte Liste  
<hr width = "300" size = "3" align = "center"> Trennlinie  
Viele Attribute, zum Beispiel align = "right" können auch in andere  
Tags eingebaut werden (mehr dazu später)
```

2.4 Tabellen

Tabellen sind in HTML ein zentrales Element. Sie können Tabellen einerseits als ganz normale Tabellen mit Gitterlinien brauchen oder als blinde Tabellen ohne Gitterlinien. Damit können Sie auch problemlos den Tabulator aus Word ersetzen, indem Sie die linke Spalte der Tabelle leer lassen.

2.4.1 (Normale) Tabellen mit Gitterlinien

Die ganze Tabelle wird mit einem `<table border = "1" width = "100%">` und einem `</table>` begrenzt. Das Attribut border definiert die Dicke der äussersten Linie um die Tabelle in Pixel, width die Gesamtbreite der Tabelle (Die Breite aller Spalten addiert). Wie wenn Sie auf Papier eine Tabelle zeichnen wollen, müssen Sie wissen, wie breit die einzelnen Spalten sein sollen. Dies definieren Sie im COLGROUP-Tag mit dem Befehl `<col width = "10%">` für die erste Spalte, `<col width = "60%">` für die zweite Spalte und so weiter. Die Summe aller prozentualen Breiten der einzelnen Spalten sollte natürlich der Breite der gesamten Tabelle entsprechen (meistens 100%). Nun haben Sie die Tabelle definiert. Gefüllt wird Sie mit den TR- und TD-Tags, siehe im Beispiel unten. Zwischen den ersten beiden TR-Tags (die der ersten Zeile entsprechen) können Sie zwischen TD-Tags die Inhalte der ersten Zeile/erste Spalte, erste Zeile/zweite Spalte und erste Zeile/dritte Spalte einfügen. Für weitere Zeilen fügen Sie einfach weitere TR-Tags mit TD-Tags ein.



```
Unbenannt - Editor
Datei Bearbeiten Format Ansicht ?
<html>
    <body>
        <table border="1" width="100%">
            <colgroup>
                <col width="10%">
                <col width="60%">
                <col width="30%">
            </colgroup>
            <tr>
                <td>A</td>
                <td>B</td>
                <td>C</td>
            </tr>
            <tr>
                <td>D</td>
                <td>E</td>
                <td>F</td>
            </tr>
        </table>
    </body>
</html>
```

Abbildung 2.4.1.1: Normale Tabelle mit Rand, drei Spalten und zwei Zeilen im Notepad

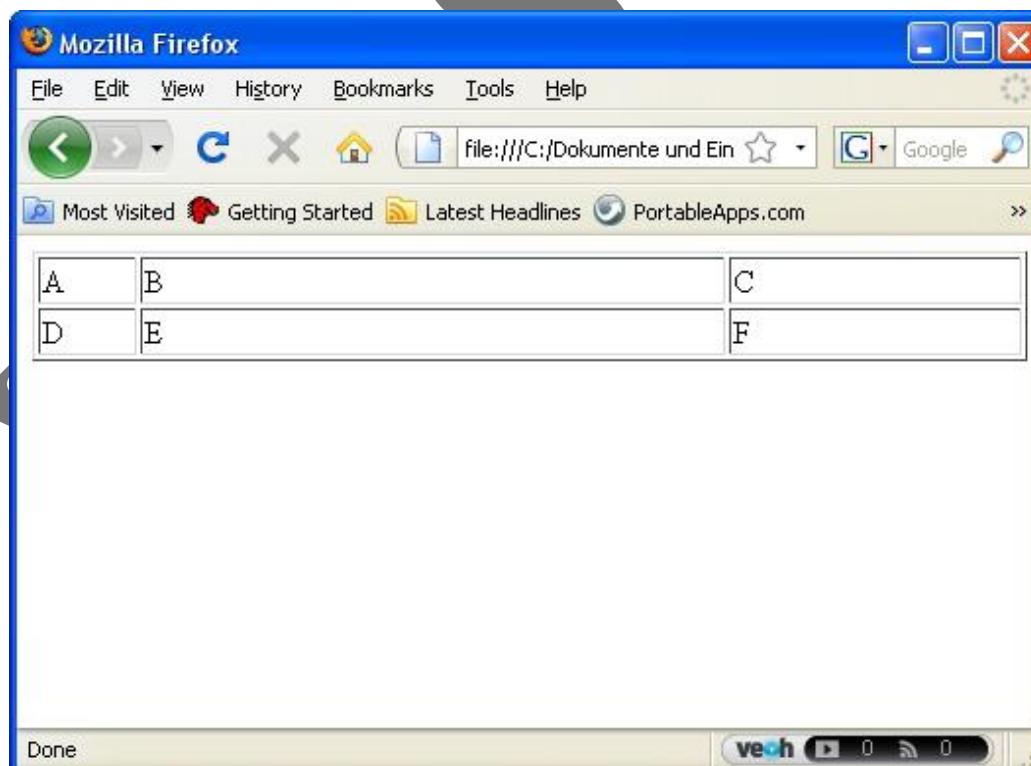


Abbildung 2.4.1.2: Normale Tabelle mit Rand, drei Spalten und zwei Zeilen im Firefox

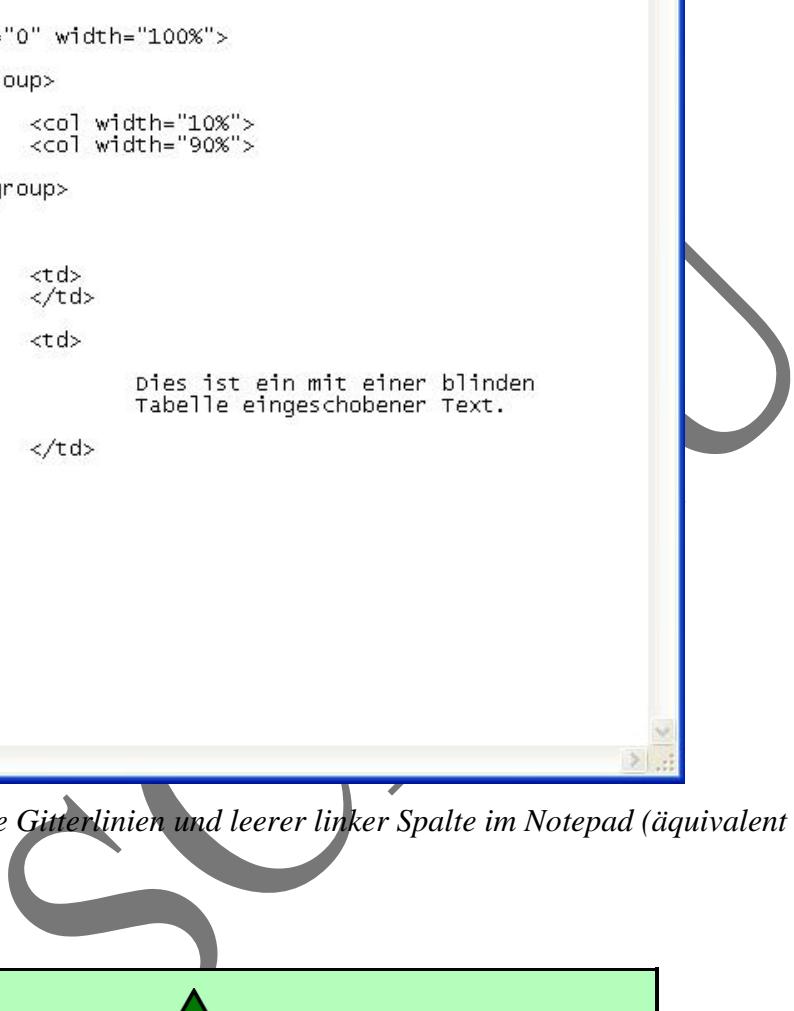
Der Vorteil von Tabellen, die auf prozentualen Angaben basieren, ist, dass die Tabelle selbst bei verkleinertem Browser (wie im Beispiel oben) noch „schön“ aussieht.



Es ist ab und zu besser, auf Tabellen zurückzugreifen, die auf absoluten Angaben (also mit Pixel) basieren. Wenn man aber dann den Browser verkleinert, muss man hin- und herscrollen, um zur gewünschten Stelle zu kommen. Um aus einer Gitterlinientabelle mit prozentualen Angaben eine Gitterlinientabelle mit Pixel-Angaben zu machen, brauchen Sie nur die Werte in den Anführungszeichen durch absolute Angaben zu ersetzen, so wird zum Beispiel width = “100%“ zu width = “1680“. Es ist im Gegensatz zur prozentualen Tabelle nicht nötig, nach dem Wert noch die Einheit anzuhängen (nach dem 100 muss noch ein % hin, nach der 1680 nichts mehr).

2.4.2 Blinde Tabellen (Tabellen ohne Gitterlinien)

Um aus einer normalen Tabelle mit Gitterlinien eine blinde Tabelle zu machen, brauchen Sie nur den border auf Null zu setzen. So wird aus `<table border = “1“ width = “100%“>` für eine normale Tabelle `<table border = “0“ width = “100%“>` für eine blinde Tabelle. Wie mit der Gitterlinien-Tabelle können Sie auch problemlos eine blinde Tabelle auf prozentualer Basis in eine entsprechende blinde Tabelle auf Pixel-Basis umwandeln, indem Sie einfach die Werte ersetzen. Wie ich schon erwähnt habe, sind blinde Tabellen zum Tabulator-Ersatz geeignet. Definieren Sie eine blinde Tabelle mit nur einer Zeile und lassen Sie die linke Spalte frei:



```
<html>
  <body>
    <table border="0" width="100%">
      <colgroup>
        <col width="10%">
        <col width="90%">
      </colgroup>
      <tr>
        <td>
        </td>
        <td>
          Dies ist ein mit einer blinden
          Tabelle eingeschobener Text.
        </td>
      </tr>
    </table>
  </body>
</html>
```

Abbildung 2.4.2.1: Tabelle ohne Gitterlinien und leerer linker Spalte im Notepad (äquivalent zum Word-Tabulator)



Tabellen sind ineinander verschachtelbar

2.4.3 Ausrichtung von Text in einer Tabelle

Definieren Sie, wie Sie es bereits kennen, eine Gitterlinien- oder eine Blinde Tabelle.

Ersetzen Sie zum Beispiel `<td>A</td>` durch `<th align = "left" valign = "top">A</th>` für die Ausrichtung von A links oben. Parallel dazu ist center top zentriert oben, right top ist rechts oben, left middle ist links mittig, center middle ist zentriert mittig, right middle ist rechts mittig, left bottom ist links unten, center bottom ist zentriert unten und right bottom ist rechts unten.

2.4.4 Färben des Hintergrunds einer Tabelle

Vielleicht wollen Sie den Hintergrund einer Zelle färben? Fügen Sie dazu in der betreffenden Zelle zum Beispiel bgcolor = "blue" zur Blaufärbung ein. So wird zum Beispiel <td>A</td> zu <td bgcolor = "blue">A</td>.

Tabellen können mit oder ohne Rand programmiert werden. Letztes ist unter Anderem als Ersatz für den Word-Tabulator geeignet.

<table border = "1" width = "100%"> Normale Tabelle mit einem Pixel dicken Rand und 100%iger Breite

<table border = "0" width = "500"> Blinde Tabelle (ohne Rand) und einer absoluten Breite von 500 Pixel.

<colgroup> Bereich zum Definieren der einzelnen Spaltenbreiten

<col width = "60%"> Diese Spalte benötigt 60% der Tabellenbreite

<col width = "30"> Diese Spalte benötigt 30 Pixel

<tr> Bereich zum Definieren der Inhalte einer Zeile

<td> Inhalt einer Spalte im Bereich einer <tr>-Zeile

<th align = "right" valign = "bottom"> Zellinhalt ist rechts unten

2.5 Links (Verweise)

Verweise sind die wichtigsten Elemente des WWW. Wenn wir zum Beispiel bei Google einen Suchbegriff eingeben, kommt, nachdem wir auf einen Link gedrückt haben (hier der Such-Button), eine neue Seite (hier mit den Suchresultaten), die wieder voller Links zu Seiten aus der ganzen Welt (daher der Name WWW = World Wide Web = Weltweites Netz) ist. Es gibt wahrscheinlich nicht eine Website auf dem Internet, die nicht in irgendeiner Form Links eingebaut hat. Daher sollten wir schleunigst lernen, wie wir in- und externe Links programmieren können.

2.5.1 Externe Links

Die wohl bekanntesten Verweise sind Links auf andere Webseiten. Einen „normalen“ Link zu zum Beispiel Google programmieren wir mit Google. Im

Browser sieht das Ganze dann so aus: [Google](#) Der Text zwischen dem Start- und Schlusstag wird blau und unterstrichen und leitet den Benutzer beim Draufklicken auf die angegebene Seite weiter.

Meistens ist es sinnvoller, wenn der Link in einem neuen Fenster geöffnet wird, damit der Benutzer nicht immer wieder auf den Zurück-Knopf im Browser drücken muss. Wenn wir zum Beispiel einen Link zu Google programmieren wollen, wobei Google in einem neuen Fenster geladen werden soll, brauchen wir das zusätzliche Attribut target = “_blank”. Für das obige Beispiel würde der vollständige Tag mit allen Attributen folgendermassen lauten: `Google`.

Es ist ab und zu schöner, wenn der Link nicht unterstrichen ist. Mit einem zusätzlichen Attribut, dem `style = “text-decoration:none;”` ist dies leicht realisierbar. Aus dem Code `Google` wird im Browser [Google](#).

Wenn wir auf einen Link klicken, nimmt der normalerweise ein Violett an. Das ist zwar ab und zu nützlich, aber nicht sehr schön. Mit den Attributen `link = “colour”`, `alink = “colour”` und `vlink = “colour”` kann die Farbe eines Links vor (Attribut `link`), während (Attribut `alink`; Abkürzung für active Link) und nach (Attribut `vlink`; Abkürzung für visited Link) dem Draufklicken definiert werden. Werden die Attribute in einen Link-Tag eingefügt (zum Beispiel `Link`), bestimmen sie nur die Farben dieses Links. Werden Sie hingegen in den BODY-Tag eingebaut (zum Beispiel `<body bgcolor = “cyan“ link = “blue“ alink = “red“ vlink = “blue“>`), sind sie für das gesamte HTML-Dokument gültig.

2.5.2 Interne Links

Interne Links sind wichtig, denn eine Website besteht meistens aus verschiedenen HTML-Dokumenten, die untereinander verlinkt sind, zum Beispiel gibt es eine Willkommensseite, eine Seite mit Text, eine mit ein paar Bildern und vielleicht noch ein paar Filme.

Interne Links programmieren wir ähnlich wie die Externen Links, die Formatierung (zum Beispiel nicht unterstrichen oder eine andere Farbe) ist genau dieselbe. Wir nehmen an, dass

das Dokument, zu dem der Link führen soll, (namens Bilder.html) im gleichen Verzeichnis (Ordner) ist wie das Dokument, in dem der Link steht. Dann schreiben wir `Bilder`. Sind die beiden Dokumente aber nicht im gleichen Verzeichnis, müssen wir nach dem href etwas leicht anderes schreiben. Wir denken uns folgende Situation für ein Beispiel aus: Auf dem Desktop befindet sich ein Dokument namens A.html und ein Ordner namens B. Im Ordner B finden wir ein Dokument namens B.html und einen Ordner namens C vor. Im Ordner C gibt es ein Dokument namens C. Unsere Aufgabe ist es, im Dokument B.html Links zum Dokument A.html und C.html anzubringen. Zusammenfassend liegt A.html relativ zu B.html in einem übergeordneten Verzeichnis und C.html in einem untergeordneten Verzeichnis.

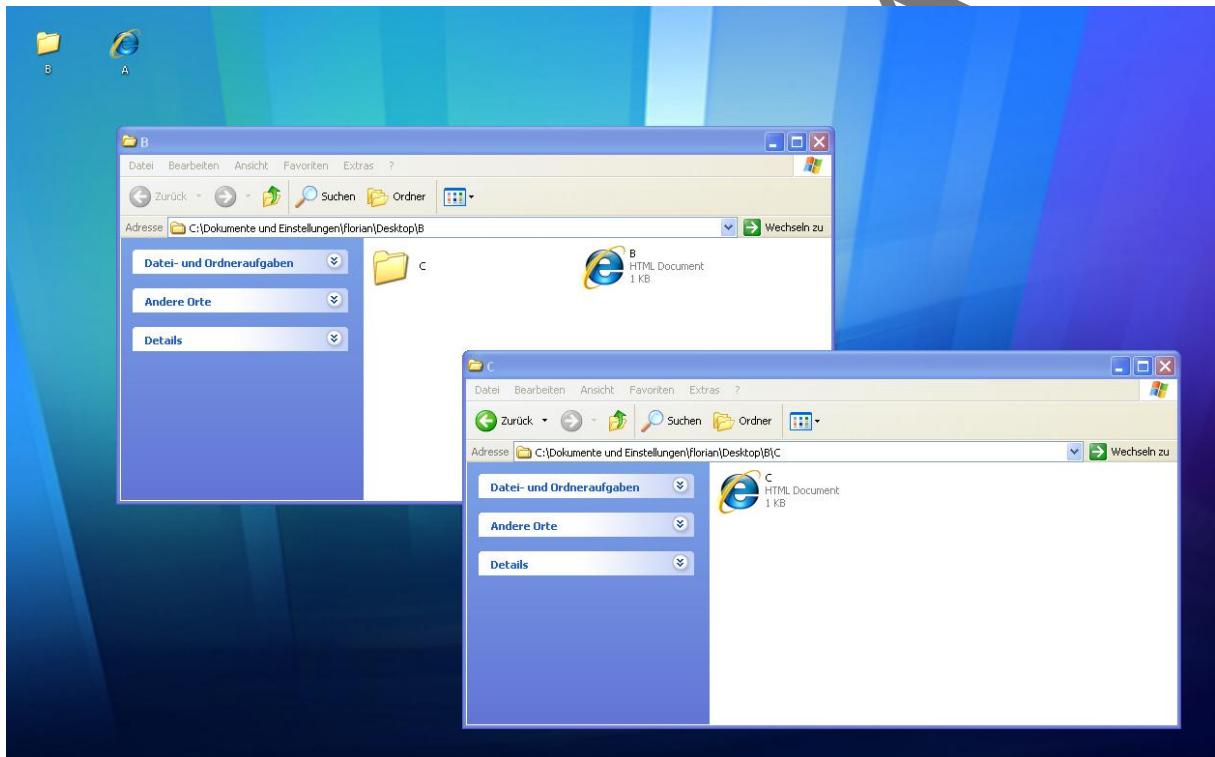


Abbildung 2.5.2.1: Grafische Darstellung des oben erläuterten Beispiels mit Betriebssystem Windows XP und Internet Explorer als Webbrowser

Zuerst der Link zum Dokument C.html: Das Dokument C.html befindet sich, wie schon erwähnt, in einem untergeordneten Verzeichnis (engl. subdirectory). Wir müssen also zuerst den Namen des Ordners (hier C) und dann das Dokument in diesem Ordner angeben: `Dokument C`. Nun zum Link zum Dokument A.html: Das Dokument A.html befindet sich in einem übergeordneten Verzeichnis (engl. parent directory, zu Deutsch in etwa elterliches Verzeichnis). Der Befehl für ein übergeordnetes Verzeichnis

lautet ../ (wie in vielen Programmiersprachen). Unser Link ist also Dokument A. Anstelle der relativen Pfadangabe (./A.html) können wir auch den absoluten Pfad angeben (C:/Dokumente und Einstellungen/Besitzer/Desktop/A.html), doch wenn wir die Dateien zum Beispiel auf ein anderes Laufwerk (D:/) kopieren, können die Dateien nicht mehr gefunden werden und man muss alle neu verlinken.

Zweites Beispiel: Wenn wir eine Datei auf dem Desktop haben (C:/Dokumente und Einstellungen/Besitzer/Desktop/) und diese mit einer Datei namens Filme.html, die direkt auf dem C: abgespeichert ist, verlinken wollen, so lautet der Befehl Filme.

2.5.3 E-Mail-Verweise

Wahrscheinlich wollen Sie auf Ihrer eigenen Webseite Ihre E-Mail-Adresse angeben. Sie können dies in Form eines Links machen, der dem Benutzer sein Standard-E-Mail-Programm (zum Beispiel Outlook) öffnet und eine neue E-Mail an den Besitzer der Webseite adressiert. Der Befehl dazu ist Mail an Franz Muster. Ein solcher Link sieht genau so aus wie ein normaler externer Link und die Formatierung ist auch dieselbe. Neben dem href = “mailto:franz.muster@hotmail.com“ – Attribut sind natürlich auch andere Link-Attribute (z.B. target = “_blank“) in den Tag einbaubar. Das spezielle an dem href-Attribut ist, dass man in ihm weitere „Unter-Attribute“ definieren kann. Das zweite Unter-Attribut ist vom ersten mit einem ? abzugrenzen, alle weiteren von den vorherigen mit einem &. Gängige Unter-Attribute vom href-Attribut sind neben dem bereits gelernten mailto:franz.muster@hotmail.com (für die E-Mail-Adresse des Empfängers) subject=Rückmeldung (für den Betreff des E-Mails) und body=Hallo, hier meine Rückmeldung zu deiner Website (für einen vordefinierten Text). Hier ein Beispiel für einen sehr komplexen E-Mail-Verweis:



Abbildung 2.5.3.1: Komplexer E-Mail-Verweis mit einem href-Attribut (unterteilt in mailto, subject und body) und einem style-Attribut in einem Link-Tag

Sie können alternativ auch E-Mail-Formulare programmieren, doch darauf werde ich in diesem Tutorial nicht näher eingehen.

2.5.4 Download-Verweise

Es gibt in HTML keinen speziellen Befehl, um dem Browser klar zu machen, dass es sich bei einer Datei um einen Download handelt. Wie in den internen Verweisen müssen Sie einfach einen Link zu der entsprechenden Datei einfügen: `Setup`. Da der Browser - im Gegenteil zu einer .html-Datei - eine .exe-Datei nicht öffnen kann, öffnet er das Download-Fenster:



Abbildung 2.5.4.1: Dateidownload-Fenster im Firefox

2.5.5 Tastenkürzel für Links

Es gibt eine Möglichkeit in HTML, für Links Tastenkürzel einzuprogrammieren: `Google`. Wenn der Benutzer nun Alt + g drückt und mit der Enter-Taste bestätigt, öffnet der Browser den angegebenen Link.



In Safari müssen Sie anstelle der Alt- die Ctrl-Taste drücken!

Einen externen Link auf eine Website wird durch den Tag `` definiert. Wichtige Attribute im Tag:

`target = "_blank"` Link wird in neuem Fenster geöffnet
`style = "text-decoration:none;"` Link ist nicht unterstrichen
`link = "blue"` Farbe des Links vor dem Draufklicken
`alink = "blue"` Farbe des Links während des Draufklickens
`vlink = "blue"` Farbe des Links nach dem Draufklicken
`accesskey = "g"` Tastenkürzel (hier Alt + g)

Einen internen Link auf ein HTML-Dokument wird durch den Tag `` definiert. Die Endung (hier .html) darf dabei nicht vergessen werden! Attribute: Siehe Oben.

Link zu einer „höheren“ Datei:

``

Link zu einer „tieferen“ Datei:

``

`` E-Mail-Verweis

Weitere Unter-Attribute: Siehe 2.5.3

2.6 Grafiken

Machen Sie auch Gebrauch von Grafiken, niemand liest gerne einen dreiseitigen Text! Laden Sie aber nicht allzu grosse Bilder hoch, man will schliesslich auch nicht eine halbe Minute warten, bis man den Text lesen kann. Eine gute Lösung sind Icons (kleine Bilder, etwa 100 auf 100 Pixel). Laden Sie grundsätzlich nur JPG-Bilder hoch, notfalls auch PNG und GIF.

Grund für JPG ist die Komprimierung des Bildmaterials in diesem Format (hohe Bildqualität bei kleiner Dateigrösse mit daraus resultierender schneller Ladezeit). Grafiken einzufügen, ist sehr einfach, Sie brauchen dazu nur den IMG-Tag: ``. Die Abkürzung img src bedeutet Image Source, also die Quelle vom Bild. Wie in den Internen Links können bzw. müssen Sie die Verzeichnisse gegebenenfalls wechseln. alt steht für

alternative, also auf Deutsch Alternativ. Alternativ insofern, als dass dieser Text solange angezeigt wird, wie das Bild geladen wird.

2.6.1 Grösse der Grafik

Vielleicht haben Sie im letzten Urlaub ein Foto mit Ihrer neuen Kamera aufgenommen, die zehn Megapixel Auflösung hat, und Sie wollen dieses Foto auf Ihrer Webseite veröffentlichen. Es sollte klar sein, dass man Ihre Site nicht mehr besuchen wird, wenn man zuerst eine halbe Minute auf ein Bild warten muss. Eine Lösung wäre das Bild verkleinert anzeigen zu lassen und einen Downloadlink unter das Bild zu setzen. Die Grösse des Bildes kann man mit zwei weiteren Attributen ganz einfach verkleinern, nämlich mit `width = "1680"` und `height = "1050"`. Da dies absolute Angaben in Pixel sind (in Prozent kann man dies nicht angeben), ist es wichtig, dass die Masse des verkleinerten Bildes proportional zu den Massen des Originals sind, weil das Bild sonst verzerrt dargestellt werden würde. Wenn wir zum Beispiel ein Original haben, das 5000 auf 2500 Pixel gross ist, können wir im Quelltext `width = "200"` `height = "100"` angeben. Die Grösse eines Bildes kann man unter den Eigenschaften (Rechtsklick auf die Grafik, Eigenschaften), Dateiinfo, Erweitert nachschauen.

2.6.2 Rahmen

Es ist auch möglich, um Bilder einen Rahmen zu machen. Sie erraten es bereits, man braucht dazu nur ein zusätzliches Attribut: `border = "5"`. Die Angabe Fünf ist in Pixel. Wollen Sie zum Beispiel um ein Bild einen 10 Pixel dicken Rahmen legen, kann der Befehl `` lauten. Die Rahmenfarbe ist diejenige, die ein Text an dieser Stelle haben würde (siehe Textformatierung).

2.6.3 Grafiken als Verweise

Sicher erahnen Sie bereits, wie man eine Grafik als Verweis definiert: ``. Wichtig ist das Attribut `border = "0"`, denn ohne diesen Befehl würde der Browser einen hässlichen Rahmen um die Grafik machen, der so aber unterdrückt werden kann.

2.6.4 Buttons als Verweise

Sie können auch Buttons als In- und Externe Verweise mit dem INPUT-Tag programmieren:

<input type = "button" value = "Hier Klicken" onclick = "location = 'http://www.google.ch'">

oder <input type = "button" value = "Hier Klicken" onclick = "location = 'index.html'">.

Tag zum Einfügen von Grafiken (im gleichen Ordner wie das

HTML-Dokument): Attribute:

alt = "Bild" Alternativer Text, bis das Bild geladen ist

width = "1680" Erzwungene Breite des Bildes in Pixel

height = "1050" Erzwungene Höhe des Bildes in Pixel

border = "5" Rahmen um das Bildes in Pixel (Rahmenfarbe

entspricht der Textfarbe an dieser Stelle (siehe Textformatierung))

Button als Verweis: <input type = "button" value = "Hier Klicken"

onclick = "location = 'http://www.google.ch'">

2.7 Multimediaobjekte

2.7.1 Videos

Fügen Sie nur Videos in Ihre Webseite ein, wenn Sie sich sicher sind, dass Sie keine Copyrights verletzen. Ihre eigenen Videos dürfen Sie natürlich in jedem Falle einbinden. Dazu braucht man den EMBED-Tag (engl. einbinden): <embed src = "Film.mpg" width = "840" height = "525">. Ändern Sie gegebenenfalls die Endung des Films (vielleicht haben Sie eine .avi-Datei) und den Pfad. Um das Video nicht zu verzieren, geben Sie die Originalgrößen an (siehe Grafiken). Sie können auch die zusätzlichen Attribute autostart = "true" (das Video startet sofort bei Seitenaufruf) und loop = "true" (das Video startet nach dem Ende neu) hinzufügen. Sie werden gelegentlich auch auf den folgenden Code stossen: <object data = "Film.mpg" width = "840" height = "525"><param name = "src" value = "Film.mpg"></object>, er bewirkt (bei äquivalenten Attributen) genau das Gleiche wie der EMBED-Tag.

2.7.2 Musik

Auch bei der Musik - oder vor Allem bei der Musik - sollten Sie sich vergewissern, dass Sie keine Copyrights verletzen. Man benutzt auch hier den EMBED-Tag: `<embed src = "Musik.mid" width = "840" height = "525">`. Wollen Sie eine Hintergrundmusik (man hört zwar die Musik, sieht aber keinen Player)? Setzen Sie die Werte nach width und height einfach auf Null! Wie bei den Videos haben Sie die Attribute `autostart = "true"` oder `autostart = "false"` und `loop = "true"` oder `loop = "false"` zur Verfügung. Sie werden auch in Bezug auf Musik auf den OBJECT-Tag stoßen.

Stellen Sie sicher, dass Sie keine Urheberrechte verletzen, bevor Sie eine Datei hochladen. Tag zum Einbinden von Video- und

Musikdateien: `<embed src = "Datei.avi">` Attribute:

`width = "840"` Breite des Filmes bzw. Musikplayers

`height = "525"` Höhe des Filmes bzw. Musikplayers

`autostart = "true"` Automatischer Start der Datei

`loop = "true"` Wiederholen des Filmes bzw. Songs

VORS

3. Upload

Wenn Sie dieses Tutorial bis hier durchgearbeitet haben, wollen Sie sicher auch eigene Daten (zum Beispiel Ihre Fotos vom letzten Urlaub) uploaden (hochladen), damit sie jede Person auf der ganzen Welt mit einem Computer und Internetanschluss ansehen können. Dazu müssen wir uns zuerst auf dem WWW einen Platz sichern, den so genannten Webspace, wo wir die Daten hochladen können. Dies ist mit einem Ordner vergleichbar (wie zum Beispiel der Ordner „Eigene Dateien“), wo wir die Dateien hineinkopieren können, nur eben online. Da wir selber keinen solchen Ordner erstellen können, brauchen wir einen Anbieter, der uns einen Ordner (den Webspace) zur Verfügung stellt.

Um die einzelnen Schritte verständlicher zu machen, werde ich sie an meinem eigenen Beispiel, <http://htmlunterricht.ht.funkpic.de> bzw. www.html-tutorial.ch.vu, wo ich auch diese Arbeit hochladen werde, erklären. Die unterschiedlichen Namen sind durch Nicht-Verfügbarkeit von html-tutorial.de erklärbar.

3.1 Webspace reservieren

Es gibt viele Anbieter, die Benutzern einen Webspace zur Verfügung stellen, einige sogar gratis (bei denen werden Sie wohl etwas Werbung in Kauf nehmen müssen). In diesem Tutorial werde ich die kostenpflichtigen Anbieter nur am Rande erwähnen. Von den gebührenfreien Anbietern hat sich für mich Funic bewährt. Gehen Sie auf www.funkpic.de und registrieren Sie sich (Hyperlink unter dem Login-Feld). Geben Sie den gewünschten Username (Benutzername) ein. Ihre Website wird in der Form <http://gewünschterUsername.ge.funkpic.de> beziehungsweise www.gewünschterUsername.ch.vu (mehr dazu später) sein. In meinem Fall muss ich also im Feld „Gewünschter Username“ „htmlunterricht“ eingeben. Es empfiehlt sich übrigens (das gilt im gesamten Informatik-Bereich), keine Umlaute und keine Leerzeichen zu verwenden (hier ist es sogar verboten). Wenn Hans Müller also eine Website erstellen will, würde seine Website www.hansmueller.ch lauten. Wie weiter unten auf der Site erläutert, kann der Username nicht mehr nachträglich geändert werden, also lieber ein zweites Mal überlegen als einen unpassenden Username registrieren!

Klicken Sie dann auf den „Weiter zu Schritt 2“-Button, um zu überprüfen, ob der Benutzername überhaupt verfügbar ist. Falls nicht, müssen Sie ihn ändern, was allerdings wegen den begrenzten Möglichkeiten (keine Sonderzeichen, also auch keine Punkte oder Kommas und keine Leerzeichen) schnell problematisch wird.

Auf der nächsten Seite müssen private Daten wie zum Beispiel eine gültige E-Mail-Adresse und ein Passwort eingegeben werden. Wenn alles richtig funktioniert hat, wurde eine Bestätigungs-Mail an die angegebene Mail-Adresse geschickt (das kann einige Minuten dauern). Nach dem Öffnen des Links im Mail wird das funpic.de-Konto aktiviert. Gehen Sie nun auf die Homepage (www.funpic.de) und loggen Sie sich mit dem Username und Ihrem Passwort ein.

Nun haben Sie Ihren persönlichen Webspace auf dem WWW reserviert. Ihre Homepage lautet <http://username.us.funpic.de>. Das „us“ nach dem Username ist variabel und entspricht den ersten beiden Zeichen des Benutzernamens. Meine Homepage lautet demnach <http://htmlunterricht.ht.funpic.de>. Es wird schnell klar, dass es nervig ist, wenn man immer eine so lange und ungewöhnliche URL eingeben muss, um die Homepage aufrufen zu können. Eine Adresse der Form www.username.ch wäre doch bedeutend einfacher. Auf switch.ch können Sie eine solche Domain (URL) registrieren lassen, was allerdings 17 Franken pro Jahr kostet. Für den privaten Gebrauch eignet sich eine Adresse der Form www.username.ch.vu; da dies gratis ist, müssen Sie ein wenig Werbung in Kauf nehmen. Diese URL fungiert wie eine Weiterleitung: Wenn jemand ins Eingabefeld des Browsers www.username.ch.vu eingibt, sendet dies automatisch eine Information an die Server von nic.ch.vu (Die Anbieter der .vu-Erweiterung) und leitet den Benutzer automatisch auf <http://username.us.funpic.de> weiter. Es können auch weitere Einstellungen wie zum Beispiel das Aktivieren von sogenannten favicons (mehr dazu später) direkt auf nic.ch.vu gemacht werden.

3.2 Umleitung reservieren

Obwohl eine Umleitung freiwillig ist, empfiehlt es sich aus oben genannten Gründen doch sehr, sich kurz Zeit dafür zu nehmen. Es gibt diverse Umleitungen, doch nic.ch.vu hat sich am Besten bewährt. Um eine Umleitung von www.username.ch.vu auf <http://username.us.funpic.de> vorzunehmen, gehen Sie wie folgt vor:

Auf der Homepage von nic.ch.vu (www.nic.ch.vu) können Sie per Link „Ist Ihr Name noch frei?“ überprüfen, ob ihr gewünschter Name noch verfügbar ist. Anders als auf funpic.de können Sie hier auch Punkte und Bindestriche verwenden, was sehr nützlich ist. Wenn Sie mit Ihrem Namen zufrieden sind und er verfügbar ist, geben Sie eine E-Mail-Adresse an (am Besten die gleiche wie auf funpic.de) und bestätigen Sie mit „weiter“. Geben Sie ein Passwort an (am Besten das gleiche wie auf funpic.de), wählen Sie den Newsletter an oder ab und schliessen Sie die Registrierung ab.

Sie können sich nun mit ihren Daten auf der Homepage von nic.ch.vu einloggen. Geben Sie Ihre Domain in der Form domain.ch.vu (bei mir also html-tutorial.ch.vu) zusammen mit Ihrem Passwort ein, um den Administrationsbereich betreten zu können. Wie Sie sofort sehen können, sind sehr viele Einstellungen verfügbar, von denen wir zuerst nur eine brauchen, nämlich die Umleitung von www.domain.ch.vu auf <http://username.us.funpic.de>. In meinem Fall muss ich also eine Weiterleitung von www.html-tutorial.ch.vu auf <http://htmlunterricht.ht.funpic.de> vornehmen. Folgen Sie dazu dem Link „Einstellungen für die Domain“ direkt unter den Kopfzeilen. Sie finden folgendes Bild vor:

Einstellungen für die Domain

Einstellungen für die Domain einsehen/ändern.

Hilfe zu den einzelnen Feldern erhalten Sie, indem Sie den Mauszeiger über die Bezeichnung des Formularfeldes bewegen.

Domain:	<input type="text" value="html-tutorial.ch.vu"/> <small>Standard</small>
Ziel-URL:	<input type="text" value="http://htmlunterricht.ht.funpic.de/index.html"/> <small>[Test]</small>
Cloaking:	<input checked="" type="radio"/> In der Browserzeile die Domain anzeigen. <input type="radio"/> In der Browserzeile die Ziel-URL anzeigen.
favicon-URL:	<input type="text" value="http://htmlunterricht.ht.funpic.de/favicon.ico"/> <small>[Auswahl]</small>
Kategorie:	<input type="text" value="Computer & Internet"/> <small>▼</small>
Titel:	<input type="text" value="HTML- Tutorial"/>
Beschreibung:	<input type="text" value="HTML-Tutorial"/>
Schlüsselbegriffe:	<input type="text" value="HTML, Tutorial, Anleitung, Website, Homepage, pr..."/>
Werbung:	<input type="text" value="keine"/> <small>▼</small> <input type="text" value="Hintergrund:"/>
 <small>» Angaben übernehmen »</small> <small>» Reset »</small>	

Abbildung 3.2.1: Einstellungen für die Domain auf nic.ch.vu

Domain: Dies ist Ihre Domain, wie sie der Benutzer eingeben soll. In meinem Fall ist dies www.html-tutorial.ch.vu.

Ziel-URL: Der Server von nic.ch.vu muss natürlich wissen, auf welche Website er den Benutzer weiterleiten soll, wenn er die Domain eingibt. Geben Sie also in diesem Feld die von funpic.de erhaltene URL ein (<http://username.us.funpic.de>). Da, wie wir bereits gelernt haben, [funpic](http://funpic.de) als eine Art Ordner fungiert, müssen wir des Weiteren noch die Datei im Ordner angeben, die wir als Homepage (Startseite) wollen. Da in der Informatik heutzutage fast alles auf Englisch ist, empfiehlt es sich, diese Datei [index.html](http://username.us.funpic.de/index.html) zu nennen. Wenn wir diese Datei einmal erstellt haben, sie auf [funpic](http://funpic.de) als „[index.html](http://username.us.funpic.de/index.html)“ in den Ordner hochgeladen haben und wir als Benutzer auf die Seite www.domain.ch.vu zugreifen wollen, werden wir automatisch auf die Datei <http://username.us.funpic.de/index.html> weitergeleitet. Im „Test“ können Sie die „Ziel-URL“ auf ihre Richtigkeit überprüfen.

Cloaking: Sie können einstellen, ob Sie in der Browserzeile (wo Sie die URL eingeben) die Domain oder die Ziel-URL anzeigen lassen wollen. Im ersten Fall steht in der Browserzeile immer <http://www.domain.ch.vu>, auch wenn Sie eine andere Datei als „[index.html](http://username.us.funpic.de/index.html)“ aufrufen wollen. Im zweiten Fall steht in der Browserzeile nicht <http://www.domain.ch.vu> sondern der tatsächliche Pfad der Datei, zum Beispiel <http://username.us.funpic.de/links.html>.

Favicon-URL: Ein Favicon ist ein kleines Icon, das links von der Browserzeile und in den Lesezeichen angezeigt wird. Fast jede professionelle grosse Seite hat ein Favicon, bei Google ist es ein weisses g auf blauem, rotem, gelbem und grünem Hintergrund, bei wikipedia ein schwarzes grosses w auf weissem Hintergrund.

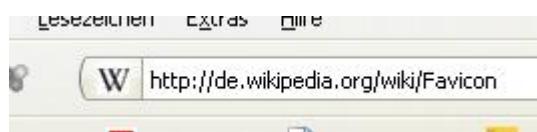


Abbildung 3.2.2: Favicon links von der Browserzeile mit der URL im Mozilla Firefox



Abbildung 3.2.3: Favicon in einem Tab links vom Titel der Site im Mozilla Firefox

Obwohl Favicons freiwillig sind, empfehle ich sie Ihnen, denn sie machen einen professionellen Eindruck und sind schnell erstellt. Leider ist es nicht möglich, ein „normales“ Bild mit der Endung .jpg zu verwenden (wäre ja zu einfach gewesen...), man muss es zuerst in eine Datei mit der Endung .ico konvertieren (umwandeln). Gehen Sie dazu auf <http://www.favicon-generator.de/>, klicken Sie auf „Durchsuchen“, wählen Sie Ihr gewünschtes Bild aus und drücken Sie auf „Hochladen“. Da Favicons immer die Grösse 16 auf 16 Pixel haben, empfiehlt es sich, das Foto zuerst quadratisch zuzuschneiden (zum Beispiel mit Adobe Photoshop), um Verzerrungen zu vermeiden. Laden Sie dann das Favicon mit dem bereitgestellten Link als „favicon.ico“ auf Ihren Computer herunter.

Sie können auf der Website auch ein Favicon mit den bereitgestellten Farben zeichnen und mit „OK“ konvertieren lassen.

Wenn wir im nächsten Kapitel zum eigentlichen Upload kommen, werden wir das favicon im Ordner <http://username.us.funpic.de> als favicon.ico speichern, in meinem Fall ist die Favicon-URL also
<http://htmlunterricht.ht.funpic.de/favicon.ico>.

Wenn Sie auf ein Favicon verzichten wollen, lassen Sie diese Zeile im Formular frei.

Kategorie: Wenn Sie wollen, können Sie die Kategorie angeben, in die Ihre Website am besten passt.

Titel: Geben Sie einen passenden Titel für Ihre Website an, der dann auf ihrer gesamten Website angezeigt wird (bei index.html, bei links.html...). Um auf ihren im HEAD-Tag definierten Titel zurückzugreifen, lassen Sie diese Zeile frei.

Beschreibung: Geben Sie eine passende Beschreibung für Ihre Website an.

Schlüsselbeg.: Geben Sie Schlüsselbegriffe ein, um es den Suchmaschinen (wie zum Beispiel Google) zu ermöglichen, Ihre Website unter den Suchresultaten auflisten zu können.

Werbung: Aktivieren oder deaktivieren Sie Werbung.

Hintergrund: Sie können eine Hintergrundfarbe für Ihre gesamte Website auswählen. Um auf die im BODY-Tag definierte Hintergrundfarbe zurückzugreifen, lassen Sie dieses Feld frei (empfohlen).

Speichern Sie die Angaben mit einem Klick auf „Angaben übernehmen“. Ab sofort ist Ihre Umleitung eingestellt und ein Benutzer wird von www.domain.ch.vu sofort auf <http://username.us.funpic.de/index.html> weitergeleitet. Nun kommen wir zum letzten und wichtigsten Teil des Uploads: dem eigentlichen Hochladen Ihrer Dateien auf den Webspace (den Ordner) auf dem Funpic.de-Server.

3.3 Dateien uploaden

Loggen Sie sich wieder auf funpic.de ein und klicken Sie auf „Webhosting FTP“. Klicken Sie bei „FTP-Status“ auf den Button „Aktivieren“. Geben Sie ein neues Passwort für den ftp-Server ein (es darf nicht das Gleiche wie für funpic.de sein) und bestätigen Sie mit „OK“. Sie können sich nun schon wieder aus dem funpic.de-Administrationsbereich ausloggen.

Mit dem Aktivieren des FTP-Servers haben Sie sozusagen den Ordner freigeschaltet, sodass Sie nun Ihre Dateien auf den Webspace hochladen können.

Obwohl Ihr Webspace unter der URL <ftp://username.us.funpic.de/> nach der Eingabe ihres Benutzernamen (von funpic.de) und ihres Passwortes (vom ftp-Server) verfügbar ist,

empfiehlt es sich, für das Hochladen von Dateien einen sogenannten FTP-Client (ein Programm) zu verwenden.

Wenn Sie mit Firefox arbeiten, empfehle ich Ihnen das Add-On „FireFTP“, das Sie unter <https://addons.mozilla.org/de/firefox/addon/684> herunterladen können. Nach dem Download müssen Sie Firefox neu starten, um das Add-On anwenden zu können. Wenn Sie nicht mit Firefox surfen, empfehle ich Ihnen, Firefox herunterzuladen (Downloadlink unter <http://www.mozilla-europe.org/de/firefox/>) oder einen anderen Client zu suchen. Wenn Sie Firefox neu gestartet haben, starten Sie den FTP-Client über Extras, FireFTP.

Erstellen Sie mit der Schaltfläche „Create an account“ in der Liste oben links einen neuen Account. Geben Sie bei „Host“ „ftp://username.us.funpic.de“ ein, bei „Login“ ihren Benutzername (username) und bei „Password“ Ihr FTP-Passwort. Der „Account Name“ dient der Übersicht in FireFTP, wenn Sie zum Beispiel mehrere Homepages betreiben wollen. Ich zum Beispiel muss bei „Host“ „ftp://htmlunterricht.ht.funpic.de“, bei „Login“ „htmlunterricht“ und bei „Account Name“ zum Beispiel „HTML-Tutorial“ eingeben. Nachdem Sie den Account angelegt haben, können Sie sich per Schaltfläche „Connect“ mit dem FTP-Server (dem Webspace) verbinden, um Ihre Daten hochzuladen.

Laden Sie zuerst eine Testdatei (zum Beispiel ein HTML-Dokument mit dem Text „Inhalt wird in Kürze hochgeladen“) mit dem Dateinamen „index.html“ (wie wir bereits gelernt haben, sollte bzw. in unserem Fall muss die Startdatei index.html heißen) hoch. Navigieren Sie dazu mit FireFTP zur gewünschten Datei, markieren Sie sie und klicken Sie auf den grünen Pfeil nach rechts, um sie auf den Server hochzuladen. Wenn der Transfer geöffnet hat, klicken Sie auf „Disconnect“, um sich vom Server zu trennen und schließen Sie FireFTP, indem Sie den Tab schließen.

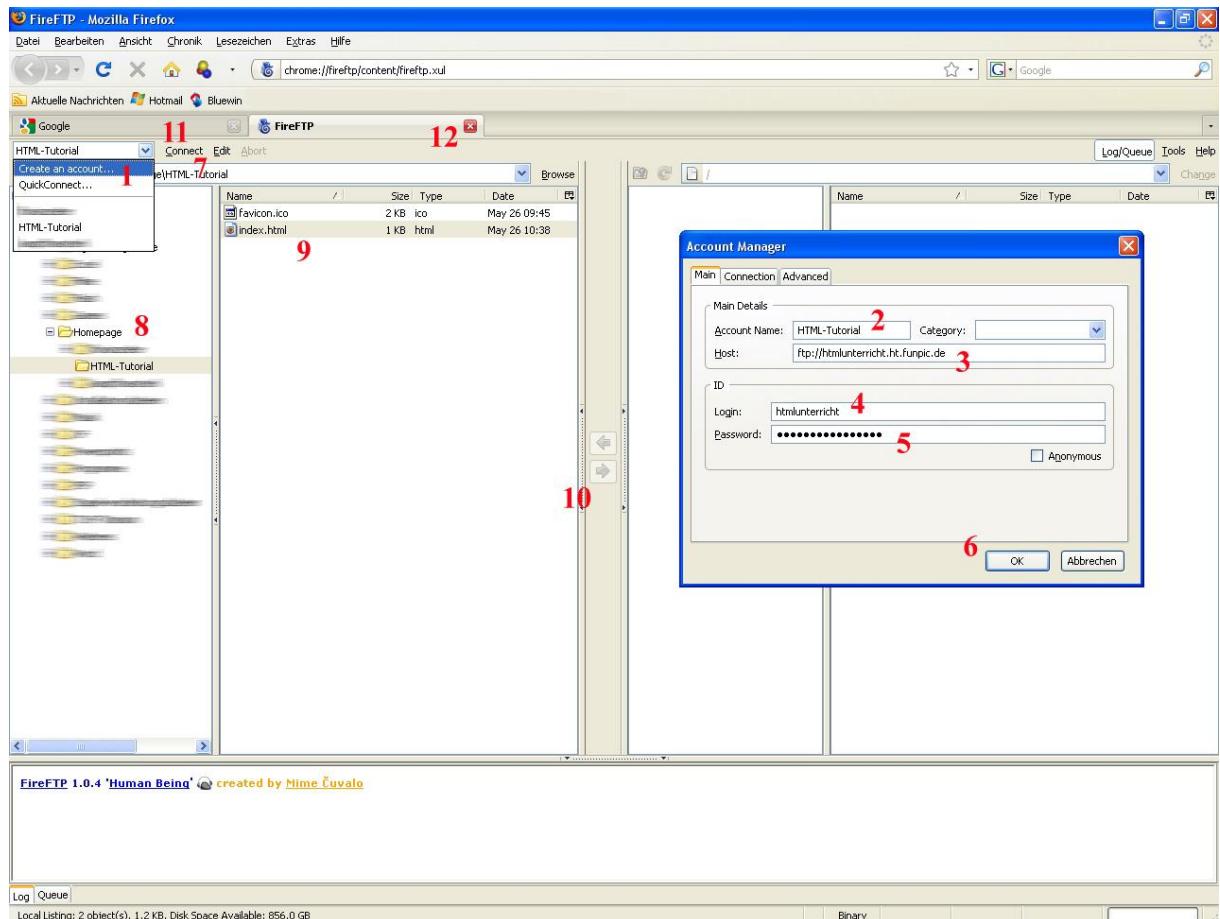


Abbildung 3.3.1: Gebrauch des Firefox-Add-Ons „FireFTP“

Wenn Sie nun auf www.domain.ch.vu oder auf <http://username.us.funpic.de> gehen, sollte im Idealfall der Text „Inhalt wird in Kürze hochgeladen“ (oder, falls Sie eine andere Testdatei erstellt haben, ein anderer Text) und Ihr Favicon (falls Sie eines erstellt und hochgeladen haben) erscheinen. Falls dies nicht der Fall sein sollte, überprüfen Sie je nach Art des Fehlers die Einstellungen auf nic.ch.vu und auf funpic.de oder kontaktieren Sie mich (html-tutorial@hotmail.com).

Erstellen Sie (falls Sie es nicht schon früher gemacht haben) einen Ordner auf Ihrem Computer, in den Sie alle Dateien (alle Bilder, alle Videos, ihr Favicon und natürlich Ihre HTML-Skripts), die Sie für Ihre Website brauchen, hineinkopieren. Wenn Sie wollen, können Sie auch Unterordner erstellen, dann müssen Sie aber gegebenenfalls die entsprechenden Links anpassen (zum Beispiel `Urlaub`).

Sobald Sie mit Ihrem Inhalt zufrieden sind, können Sie die Dateien im Ordner – wie schon die Testdatei – mit Hilfe von FireFTP auf Ihren FTP-Server hochladen.

Website gratis veröffentlichen (mit Werbung):

1. Konto auf funpic.de erstellen, Webspace lautet
<http://gewünschterUsername.ge.funpic.de>
2. Konto auf nic.ch.vu erstellen, Umleitung lautet
<http://www.gewünschterUsername.ch.vu>
3. Auf nic.ch.vu Einstellungen für die Domain vornehmen
4. Auf funpic.de FTP-Status aktivieren
5. Firefox und Add-On FireFTP downloaden
6. Dateien hochladen

Website kostenpflichtig veröffentlichen (ohne Werbung):

1. Konto auf switch.ch erstellen (17.- pro Jahr), Webspace lautet
<http://www.gewünschterUsername.ch>
2. Anbieter für Filehosting auswählen (je nach Bedürfnisse auswählen; siehe Linkssammlung auf switch.ch)

4. Weiterführende Links

w3schools (Alternatives Tutorial):

<http://www.w3schools.com/html/default.asp>

Tutorial und Nachschlagewerk:

<http://de.selfhtml.org/html/index.htm>

CSS-Tutorial (aufbauend auf HTML):

<http://de.html.net/tutorials/css>

Homepage des Tutorials:

<http://www.html-tutorial.ch.vu/>

Homepage des Autors

<http://www.florianzeiter.ch.vu/>